



Escuela
Politécnica
Superior

Control de marcha para robots humanoides que suben escaleras



Máster Universitario en Automática y Robótica

Trabajo Fin de Máster

Autor:

Alfredo Fenoll Quirant

Tutor/es:

Gabriel Jesus García Gómez

Julio 2021



Universitat d'Alacant
Universidad de Alicante

Control de marcha para robots humanoides que suben escaleras

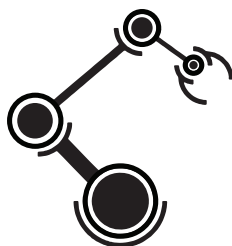
Autor

Alfredo Fenoll Quirant

Tutor/es

Gabriel Jesus García Gómez

Física, Ingeniería de Sistemas y Teoría de la Señal



Máster Universitario en Automática y Robótica



Escuela
Politécnica
Superior



Universitat d'Alacant
Universidad de Alicante

ALICANTE, Julio 2021

Justificación

El estudio de la locomoción de robots humanoides, es decir, de aquellos robots que han sido diseñados para imitar al ser humano, tanto en forma como en comportamiento, es una de las claves para lograr una integración exitosa de los robots en el entorno social. Para lograr que los robots puedan ayudar a la sociedad en muchos ámbitos, como podrían ser la asistencia a personas mayores y a enfermos, o realizando actividades de apoyo psicoterapéutico y social.

A su vez, dotar a los robots humanoides de las habilidades motoras para poder interactuar con el entorno urbana, donde realizar todas estas actividades de servicio social, supone un gran reto a nivel tecnológico.

Por ello es objetivo de este trabajo, realizar el estudio de movimiento necesario para que los robots humanoides sean capaces de subir escaleras, un elemento muy común en el entorno urbano y residencial, que a su vez presenta un gran obstáculo para este tipo de robots.

Durante este proyecto se realizará un estudio del arte de la marcha en robots humanoides y se aplicará lo aprendido sobre el robot humanoide NAO, el cual deberá aprender a subir escaleras de la forma más rápida, eficiente y segura que sea capaz. Para ello deberemos estudiar diferentes ámbitos de la ciencia como:

- Estudio de la evolución de la robótica humanoide bípeda hasta nuestros días.
- El estudio biomecánico de la marcha humana.
- Estrategias de generación patrones de marcha en robots.

Agradecimientos

En primer lugar, quisiera agradecer a mis padres, por haberme inspirado y respaldado durante toda mi educación. A mi hermano, por apoyarme en los momentos difíciles y a mi abuelo Alfredo, por darme una perspectiva diferente de la vida, la de una persona que ha basado su vida en la cultura del esfuerzo y la humildad.

También quiero agradecerle a mi tutor, Gabriel, por haberme guiado y ayudado, haciendo que el trabajo a realizar, que al principio parecía irrealizable, se transformase en una experiencia interesante e instructiva.

Sin olvidarme de la Universidad de Alicante y a todo su profesorado y demás trabajadores del campus, en especial al grupo de investigación Human Robotics, que nos han apoyado durante la realización de todo el máster y nos han brindado todo su apoyo y conocimientos. A todos los compañeros que me han acompañado durante este año, conocerlos a sido enriquecedor tanto a nivel personal como profesional.

Y a todas aquellas personas que, de una forma u otra, me han ayudado a lo largo de esta grata aventura que ha resultado ser mi vida en la universidad.

*A Sara,
porque siempre está a mi lado cuando la necesito,
porque ella es única haciéndome sonreír y contagiándome su vitalidad*

*”La mayoría de las personas gastan
más tiempo y energías en hablar de
los problemas que en afrontarlos”*

Henry Ford.

Índice general

1	Introducción	1
1.1	Un pequeño vistazo a la historia de la robótica humanoide	1
1.2	Historia moderna	3
2	Estado del Arte	7
2.1	Estudio de la Marcha Humana	7
2.2	Diferencias entre caminar y subir escaleras	9
2.3	Patrones de marcha en robots	12
2.3.1	Zero-Moment Point	12
2.3.1.1	Centro de Presión (CoP) y su relación con el ZMP	16
2.3.1.2	Cálculo del ZMP por cinemática directa	17
2.3.1.3	Modelo Cart-Table	19
2.3.1.4	Limitaciones del modelo ZMP	20
2.3.2	Estrategias de Marcha Alternativas al ZMP	20
2.3.2.1	Marcha Pasiva	20
2.3.2.2	Passive Dynamic Walking	22
3	Objetivos	25
4	Metodología	27
4.1	Entorno de trabajo	27
4.1.1	Robot NAO	27
4.1.1.1	Características técnicas	27
4.1.2	Software de desarrollo	30
4.1.2.1	NAOqi OS	30
4.1.2.2	Choreographe	30
4.1.2.3	Monitor	33
4.2	Desarrollo	34
4.3	Estrategia propuesta para subir escaleras	35
5	Código Implementado	41
6	Resultados	47
7	Conclusiones	57
7.1	Trabajos futuros	58
	Bibliografía	59
	Lista de Acrónimos y Abreviaturas	63

Índice de figuras

1.1	Doríforo de Policleteo. Fuente: https://grupenciclopedia.cat/blog/es/el-doriforo-de-policleteo/	1
1.2	Aportaciones de Leonardo Da Vinci al mundo de la anatomía y la robótica	2
1.3	Robot Wabot 1. Fuente: https://www.abadiadigital.com/wp-content/uploads/2014/11/wabot-11-412x495.jpg	3
1.4	Robot Wabot 2. Fuente: https://www.abadiadigital.com/wp-content/uploads/2014/11/wabot-11-412x495.jpg	4
1.5	Robot Asimo subiendo una escalera. Fuente: https://clipset.com/asimo-el-robot-de-honda/	4
1.6	Robot NAO V6. Fuente: https://www.softbankrobotics.com/emea/es/nao	5
1.7	Robots Nao durante la final de la RoboCup 2010. Fuente: https://www.researchgate.net/figure/On-the-left-a-NAO-robot-used-in-the-SPL-competitions-On-the-right-an-image-from-the	5
1.8	Robot Atlas saltando. Fuente: https://spectrum.ieee.org/robotics/humanoids/how-boston-dynamics-is-redefining-robot-agility	6
2.1	Las cuatro fases del apoyo durante la marcha. Fuente: https://www.elsevier.es/es-revista-revista-iberoamericana-fisioterapia-kinesiologia-176-articulo-fases-marcha-humana-13012714	8
2.2	Fases de apoyo y oscilación durante la marcha. Fuente: https://www.researchgate.net/figure/Figura-1-Ciclo-de-Marcha-y-sus-Fases-3-Medicion-de-Parametros-de-la-Marcha-Humada_fig1_321492779	9
2.3	Trayectoria típica de desplazamiento del centro de gravedad de una persona durante el ciclo de marcha. Fuente: https://m.exam-10.com/pravo/6637/index.html?page=4	9
2.4	Robot WL-10RD, Fuente: http://sashrobo.blogspot.com/2015/01/shasher.html	12
2.5	Representación gráfica del concepto ZMP, Fuente: https://www.mdpi.com/2076-3417/9/3/502/htm	13
2.6	Polígono de soporte para robot bípedo. Fuente: Elaboración propia	13
2.7	Humanoide en "posición anatómica" incluyendo representación de planos y origen, Fuente: https://www.techunited.nl/media/files/humanoid/MaartenDekker_OPEN2009_Zero_Moment_Point_Method_for_Stable_Biped_Walking.pdf	14
2.8	Fuerzas y momentos actuando en un pie rígido en contacto con el suelo. Fuente: https://www.techunited.nl/media/files/humanoid/MaartenDekker_OPEN2009_Zero_Moment_Point_Method_for_Stable_Biped_Walking.pdf	15
2.9	Relación entre el ZMP y el CoP: (a) Dinámicamente estable. (b) Dinámicamente inestable. Fuente: https://www.techunited.nl/media/files/humanoid/MaartenDekker_OPEN2009_Zero_Moment_Point_Method_for_Stable_Biped_Walking.pdf	16

2.10	Modelo esquemático del robot bípedo y el punto P. Fuente: https://www.techunited.nl/media/files/humanoid/MaartenDekker_OPEN2009_Zero_Moment_Point_Method_for_Stable_Biped_Walking.pdf	17
2.11	Modelo Cart-Table. Fuente: https://www.techunited.nl/media/files/humanoid/MaartenDekker_OPEN2009_Zero_Moment_Point_Method_for_Stable_Biped_Walking.pdf	19
2.12	Dynamite, “Passive Dynamic Walker” con rodillas de McGeer. Fuente: https://intranet.ceautomatica.es/old/actividades/jornadas/XXIX/pdf/240.pdf	20
2.13	De derecha a izquierda: “Toddler” del MIT, Cornell Efficient Biped, “Denise” de la TU Delft, Fuente: https://intranet.ceautomatica.es/old/actividades/jornadas/XXIX/pdf/240.pdf	22
4.1	Dimensiones del Robot NAO	27
4.2	Vista de las articulaciones del robot NAO. Fuente: http://doc.aldebaran.com/2-8/family/nao_technical/lola/actuator_sensor_names.html	28
4.3	Workspace de Choregraphe con bloques de funciones. Fuente: http://doc.aldebaran.com/2-8/software/choregraphe/objects/flow_diagram_box.html	31
4.4	Vista del editor de parámetros para la función “Move To”. Fuente: Elaboración propia	31
4.5	Vista de la pantalla de edición de un bloque Python. Fuente: Elaboración propia	32
4.6	Código predeterminado dentro de un bloque Python. Fuente: Elaboración propia	32
4.7	Vista principal de programa Monitor. Fuente: Elaboración propia	33
4.8	Gáfica del movimiento angular de las articulaciones del robot al subir un peldaño de la escalera. Fuente: Elaboración propia	33
4.9	Robot humanoide Pepper. Fuente: https://www.softbankrobotics.com/emea/es/pepper	34
4.10	Vistas de la escalera utilizada.	35
4.11	Posición predefinida “StandInit”. Fuente: http://doc.aldebaran.com/2-8/family/nao_technical/postures_naov6.html	36
4.12	Movimiento 1º, movimiento lateral, de posición “StandInit” a posición de apoyo monopodal. Fuente: Elaboración propia	36
4.13	Croquis del movimiento 2	37
4.14	Croquis del movimiento 3	38
4.15	Croquis del movimiento 4	38
4.16	Croquis del movimientos 5	39
5.1	Sistema de referencia usado en el módulo ALMotion. Fuente: http://doc.aldebaran.com/2-8/naoqi/motion/almotion.html	42
6.1	Evolución de la posición angular con respecto del tiempo en las articulaciones del robot	47
6.1	Evolución de la posición angular con respecto del tiempo en las articulaciones del robot	48
6.1	Evolución de la posición angular con respecto del tiempo en las articulaciones del robot	49

6.1	Evolución de la posición angular con respecto del tiempo en las articulaciones del robot	50
6.2	Imágenes del robot NAO subiendo las escaleras, momentos destacados del ciclo	51
6.2	Imágenes del robot NAO subiendo las escaleras, momentos destacados del ciclo	52
6.2	Imágenes del robot NAO subiendo las escaleras, momentos destacados del ciclo	53
6.2	Imágenes del robot NAO subiendo las escaleras, momentos destacados del ciclo	54
6.3	Workflow de Choregraphe con el programa cargado. Fuente: Elaboración propia	55

Índice de tablas

2.1	Ciclo típico de una persona al subir escaleras. Fuente: Elaboración propia a partir de: https://www.researchgate.net/profile/James-Zachazewski/publication/15039814_Biomedical_analysis_of_body_mass_transfer_during_stair_ascent_and_descent_of_healthy_subjects/links/0912f511a9dcce178b000000/Biomedical-analysis-of-body-mass-transfer-during-stair-ascent-and-descent-of-healthy-subjects.pdf	10
4.1	Tabla cadenas cinemáticas y articulaciones del NAO. Fuente: Elaboración propia a partir de: http://doc.aldebaran.com/2-8/family/nao_technical/bodyparts_naov6.html	29
4.2	Table de los diferentes tipos de motores delNAO. Fuente: Elaboración propia a partir de http://doc.aldebaran.com/2-8/family/nao_technical/motors_naov6.html	29

Índice de Códigos

5.1	Código Python para hablar, NAO	41
5.2	Código Python listas vacías	41
5.3	Ejemplo del código relativo únicamente a la articulación LKneePitch (flexión de la rodilla izquierda)	41
5.4	Código utilizado para llamar al módulo "ALMotion" y al método "angleInterpolation")	41
5.5	Código utilizado para llamar al módulo "ALMotion" y al método "angleInterpolation")	42
5.6	Código en Python para el movimiento de subir escalón con la pierna derecha	43
5.7	Código en Python para el movimiento de subir escalón con la pierna izquierda	45

1 Introducción

1.1 Un pequeño vistazo a la historia de la robótica humanoide

La historia de los robots humanoides es una historia más antigua de lo que cabría imaginar, de hecho, el ser humano siempre ha estado obsesionado con el cuerpo humano, y una de sus metas siempre ha sido el llegar a imitarlo. En la antigüedad esa obsesión se canalizaba a través de la pintura o la escultura.

Un ejemplo lo tenemos en la cultura griega, que ya en el siglo IV a. C. eran capaz de representar de forma meticulosa la morfología anatómica del cuerpo humano a través de la escultura, logrando un aspecto final extremadamente realista, incluso representando las tensiones y las flexiones de los diferentes grupos musculares que conforman el cuerpo humano (*Doríforo*, Fecha de la última visita: 14/06/21), (*El Doríforo de Policleto, la escultura que marcó el canon*, Fecha de la última visita: 14/06/21).

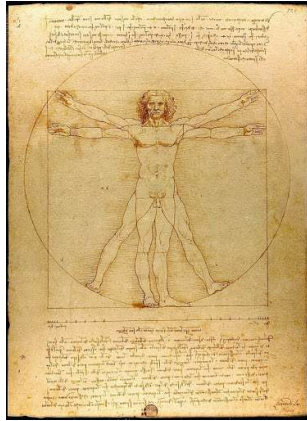


Figura 1.1: Doríforo de Policleto. Fuente: <https://grupenciclopedia.cat/blog/es/el-doriforo-de-policleto/>

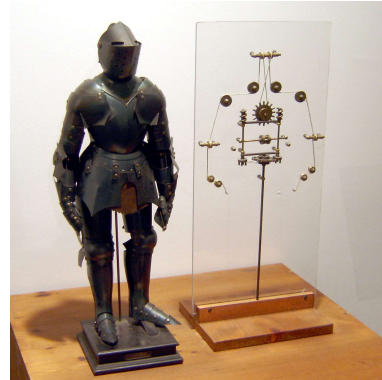
En siglos posteriores, las representaciones fueron refinándose, hasta aparecer los primeros humanoides mecánicos, como lo fue el Robot de Leonardo, que como su nombre indica, fue diseñado por Leonardo da Vinci alrededor del año 1495. Las notas de diseño que aparecen en sus cuadernos de bocetos permitieron reconstruir el dispositivo, de forma que se demostró que era plenamente funcional (*Robot de Leonardo*, Fecha de la última visita: 14/06/21).

El robot es un guerrero vestido con una armadura medieval germano-italiana, que aparentemente es capaz de hacer varios movimientos parecidos a los de los humanos. En estos movimientos se incluyen sentarse, mover los brazos, el cuello y la mandíbula de una forma anatómicamente perfecta. Se interpreta que esto es fruto de la investigación de las proporcio-

nes anatómicas que Leonardo plasmo en su famoso dibujo del "Hombre de Vitruvio" (*Hombre de Vitrubio*, Fecha de la última visita: 06/06/21).



(a) Hombre de Vitruvio. Fuente:
[http://arts.recursos.uoc.edu/
referents-dibuix/es/
leonardo-da-vinci/](http://arts.recursos.uoc.edu/referents-dibuix/es/leonardo-da-vinci/)



(b) Robot de Leonardo. Fuente:
[https://es.wikipedia.org/
wiki/Robot_de_Leonardo](https://es.wikipedia.org/wiki/Robot_de_Leonardo)

Figura 1.2: Aportaciones de Leonardo Da Vinci al mundo de la anatomía y la robótica

1.2 Historia moderna

No es hasta el siglo XX cuando se producen grandes avances en el campo de la robótica, impulsado por el notable avance que experimentan las ciencias de la computación, control sensorial e inteligencia artificial. Aunque, uno de los primeros hechos significativos que se producen en ese siglo, es que el propio término de “robot” es acuñado por Karel Čapek, en su obra *Rossum’s Universal Robots* en el año 1920. La obra era una sátira donde los robots eran seres biológicos fabricados para que realizaran todo el trabajo manual desagradable. Según Čapek, la palabra la creó su hermano Josef que en checo “robotá”, significa “servidumbre”.

En 1973 aparece el WABOT 1 (figura 1.3), el primer robot humanoide a escala real construido en el mundo. Entre sus capacidades destacan el reconocimiento de objetos y reconocimiento y generación de voz además de sus funciones de locomoción, era capaz de dar unos cuantos pasos lentos y ser estáticamente estable. El robot fue creado por un grupo de ingenieros de la Universidad Waseda de Tokio, su nombre WABOT, es un acrónimo que hace referencia a su origen (WAseda roBOT) (*WABOT, la familia de robots humanoides de la Universidad de Waseda*, Fecha de la última visita: 19/06/21).

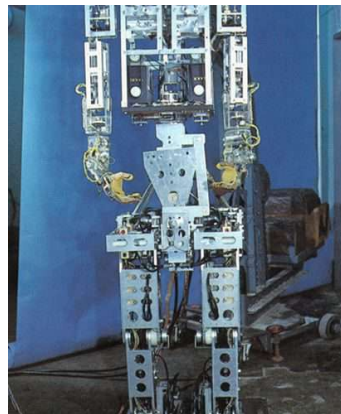


Figura 1.3: Robot Wabot 1. Fuente: <https://www.abadiadigital.com/wp-content/uploads/2014/11/wabot-11-412x495.jpg>

En 1984 vio la luz su predecesor, el WABOT-2 (figura 1.4) que contaba con un aspecto humanoide y podía recitar y entender algunas palabras en japonés, pero además era capaz de leer una partitura e incluso tocar una melodía de mediana dificultad con un sintetizador. De la misma manera que sucedió con su predecesor, en su momento fue considerado como el robot más avanzado del mundo (Humanoid Robotics Institute, Fecha de la última visita: 07/06/21).



Figura 1.4: Robot Wabot 2. Fuente: <https://www.abadiadigital.com/wp-content/uploads/2014/11/wabot-11-412x495.jpg>

En los años 80 y 90 Honda Motor Company crea un centro de investigación para desarrollar los robots humanoides de la serie P (“Prototipo”). El desarrollo empieza en 1986, con el objetivo principal de estudiar la locomoción humana, usando el esqueleto humano como modelo para crear una réplica y dotarla de un movimiento lo más parecido a la marcha humana. A partir de 1993 el estudio ya no se limitaba solo a estudiar la marcha humana, si no al cuerpo en su conjunto, por lo que se añadieron a los modelos torso, brazos y cabeza. En 1996 se lanzó el segundo prototipo, el P2, usando tecnología inalámbrica este robot se convirtió en el primer robot bípedo capaz de caminar por sí solo (*Asimo*, Fecha de la última visita: 02/06/21).



Figura 1.5: Robot Asimo subiendo una escalera. Fuente: <https://clipset.com/asimo-el-robot-de-honda/>

La serie P después de cuatro iteraciones, concluiría con la creación del robot ASIMO (figura 1.5) en el año 2000, este robot supuso toda una revolución tecnológica en el mundo de la robótica humanoide. Por primera vez, un robot bípedo, de dimensiones y formas humanas, era capaz de mantener perfectamente el equilibrio al caminar y hasta podía subir y bajar escaleras. Este increíble avance, que simulaba el sistema de equilibrio de las personas. Se convirtió en el primer robot humanoide del mundo capaz de interactuar autónomamente en

un entorno humano (*Asimo El robot humanoide más avanzado del mundo*, Fecha de la última visita: 02/06/21).

Con la entrada al siglo XXI la llegada de robots humanoides procedentes de diversas empresas de todas partes del mundo ha sido una constante. Entre ellos el robot que va a ser utilizado en este proyecto, el robot NAO (figura 1.6), desarrollado a partir del año 2004 por Aldebaran Robotics una compañía de robótica francesa subsidiaria del grupo Softbank ?.

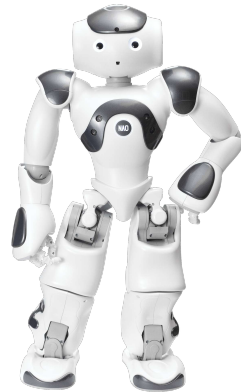


Figura 1.6: Robot NAO V6. Fuente: <https://www.softbankrobotics.com/emea/es/nao>

Este robot fue desarrollado para las universidades y laboratorios con fines de investigación y educación. Además, fue el robot elegido como plataforma oficial de la Robocup ("Robot Soccer World Cup"), un concurso internacional de robótica donde los robots deben jugar entre ellos partidos de futbol de manera autónoma y realizando movimientos que imiten al movimiento de un humano (figura 1.7) (*RoboCupSoccer - Humanoid*, Fecha de la última visita: 04/06/21).

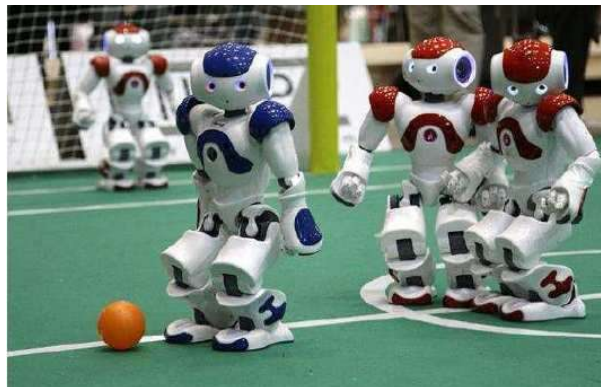


Figura 1.7: Robots Nao durante la final de la RoboCup 2010. Fuente: <https://www.researchgate.net/figure/On-the-left-a-NAO-robot-used-in-the-SPL-competitions-On-the-right-an-image-from-the>

En la actualidad uno de los robots humanoides más destacados es el robot Atlas (figura 1.8) de la compañía americana Boston Dynamics, ahora en propiedad de Hyundai Motor Company, este robot no tiene un aspecto amigable o social tan marcado como otros robots, pero eso se debe a que la interacción social no es uno de los propósitos de su diseño, el robot fue financiado por la Agencia de Proyectos de Investigación Avanzados de Defensa (DARPA), una agencia perteneciente al departamento de defensa de Estados Unidos, para ayudar a los servicios de emergencia en las operaciones de búsqueda y rescate, realizando tareas como cerrar válvulas, abrir puertas y operar equipos motorizados en entornos donde los humanos no podrían sobrevivir. Por lo cual sus capacidades motrices y el control sobre su cuerpo eran la prioridad (*Atlas*, Fecha de la última visita: 05/06/21). Los resultados son extraordinarios, en varias demostraciones se puede ver al robot realizando movimientos muy complejos, como volteretas, giros rotando 360 grados sobre sí mismo en el aire e incluso realizando pasos de baile con una naturalidad muy cercana a la del ser humano (Dynamics, Fecha de la última visita: 05/06/21).



Figura 1.8: Robot Atlas saltando. Fuente: <https://spectrum.ieee.org/robotics/humanoids/how-boston-dynamics-is-redefining-robot-agility>

2 Estado del Arte

2.1 Estudio de la Marcha Humana

El estudio de la marcha humana es de vital importancia para conocer los patrones de marcha que utilizan los humanos a la hora de subir escaleras. A lo largo de este apartado exploraremos en profundidad esta habilidad tan característica del ser humano y veremos cuáles son las principales diferencias que plantea con respecto a la estrategia que siguen los humanos a la hora de subir escaleras.

La marcha bípeda es una de las habilidades más importantes y diferenciadoras del ser humano, además de ser la característica más antigua que se conoce del ser humano. Los primeros sujetos erectos que se conocen datan ya de 4.000.000 de años (“Fases de la marcha humana”, s.f.).

La marcha bípeda o marcha humana, es un proceso de locomoción en el cual el cuerpo humano, se mueve hacia delante, siendo su peso soportado alternativamente por ambos miembros inferiores. Se caracteriza a la vez que se distingue de la carrera por el contacto permanente del individuo con el suelo a través de al menos uno de sus pies.

El ciclo de marcha bípeda podemos dividirlo en cuatro periodos en función del reparto del peso del cuerpo entre las extremidades inferiores (“Cinesiología de la marcha humana normal”, Fecha de la última visita: 08/06/21):

- Periodo de apoyo doble: Es el periodo en que ambos pies están totalmente apoyados en el suelo, comienza cuando el pie tomado como referencia, toma contacto con el suelo por el talón, frenando la aceleración del cuerpo hacia delante y culmina con el despegue del miembro contralateral.
- Primer periodo de apoyo simple: es la fase en que solo un pie está apoyado totalmente en el suelo y el otro pie se mueve hacia delante por el aire. El peso del cuerpo recae en la extremidad tomada como referencia, mientras el miembro contralateral está oscilando.
- Segundo periodo de apoyo doble: El pie de referencia se apoya solo por el antepié en el suelo y está en situación posterior acelerando el cuerpo hacia delante, el otro pie ya ha finalizado su periodo de oscilación y aterriza sobre el suelo.
- Segundo periodo de apoyo simple: El pie que en el tiempo anterior solo se apoyaba por el antepié en el suelo, ha despegado e inicia su periodo oscilante, mientras que el otro se encuentra apoyado sosteniendo todo el peso de la persona.

Como podemos intuir de la segmentación por periodos que acabamos de realizar, durante un ciclo completo de marcha, ambas extremidades inferiores pasan por dos fases, por un lado,

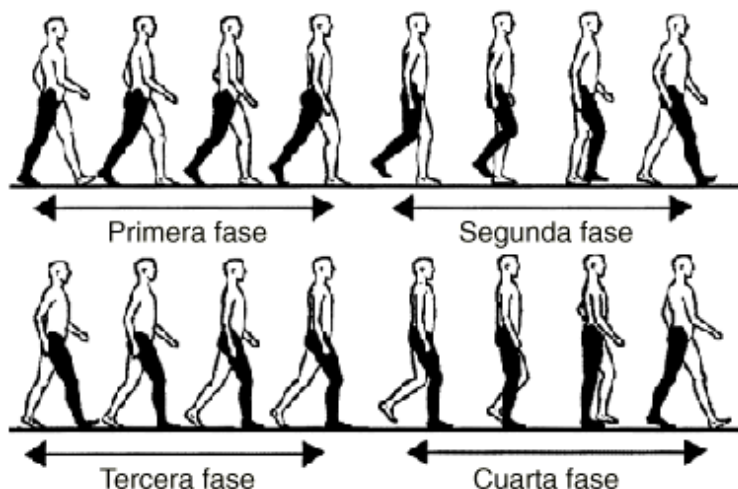


Figura 2.1: Las cuatro fases del apoyo durante la marcha. Fuente: <https://www.elsevier.es/es-revista-revista-iberoamericana-fisioterapia-kinesiologia-176-articulo-fases-marcha-humana-13012714>

tenemos la fase de apoyo, en la cual, el pie está en contacto con el suelo, esta fase se alterna con la fase de oscilación, en la cual, el pie se encuentra suspendido en el aire.

La fase de apoyo constituye alrededor del 60% del ciclo y la fase de oscilación representa el 40% restante. A su vez, tanto la fase de apoyo como la fase de oscilación pueden subdividirse en subfases. En el caso de la fase de apoyo lo dividimos en 5 subfases, que se pasan a ser descritas según su orden cronológico (“Biomecánica de la Marcha y la Carrera”, Fecha de la última visita: 08/06/21):

- Fase de Contacto Inicial: En esta fase el talón contacta con el suelo.
- Fase inicial del apoyo o de respuesta a la carga: Transcurrido aproximadamente el 15% del tiempo del ciclo total, el antepié contacta con el suelo, en esta fase tenemos el pie totalmente apoyado en el suelo.
- Fase de Soporte Medio: El comienzo de esta fase viene marcado por el despegue de dedos del miembro contralateral. En la pierna de apoyo se estabiliza la cadera y la rodilla mientras el cuerpo avanza sobre un pie estacionario.
- Fase de Soporte Terminal: Transcurrido aproximadamente el 40% del tiempo del ciclo total, el talón comienza a elevarse del suelo.
- Fase previa a la Oscilación: Esta fase culmina con el despegue de los dedos del suelo, y coincide con la fase de contacto inicial del pie opuesto.

La fase de oscilación o balanceo se divide en tres subfases que vendrían a ser las siguientes:

- Fase de Oscilación Inicial: Los objetivos básicos de esta fase son conseguir una separación pie-suelo suficiente, así como alcanzar la cadencia deseada.
- Fase de Oscilación Media

- Fase de Oscilación Terminal: En esta fase los objetivos son decelerar la pierna y pre-posicionar correctamente el pie para establecer contacto con el suelo. Es necesaria una extensión completa de la rodilla y una posición neutra del pie con respecto a la pierna para realizar el contacto efectivo del talón y el comienzo del siguiente ciclo.

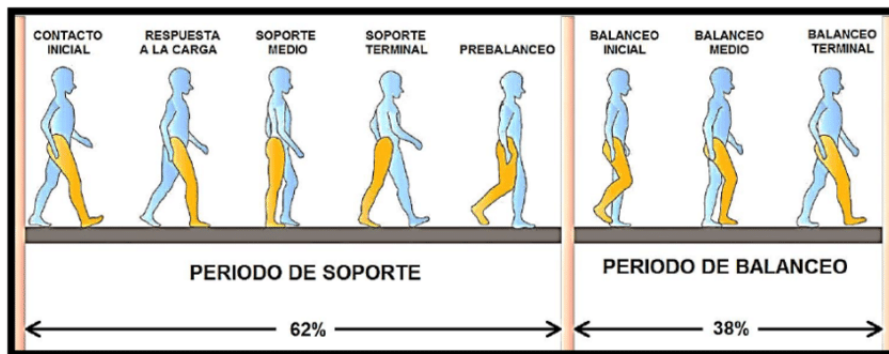


Figura 2.2: Fases de apoyo y oscilación durante la marcha. Fuente: https://www.researchgate.net/figure/Figura-1-Ciclo-de-Marcha-y-sus-Fases-3-Medicion-de-Parametros-de-la-Marcha-Humada_fig1_321492779

2.2 Diferencias entre caminar y subir escaleras

La diferencia más evidente entre el movimiento de caminar y subir escaleras es el comportamiento del centro de gravedad del cuerpo.

Durante el ciclo de marcha al caminar, el centro de gravedad sufre pequeñas oscilaciones siguiendo un patrón sinusoidal de unos 5 centímetros, de una forma natural las personas tienden a realizar patrones de movimiento que alteren lo menos posible el desplazamiento vertical del centro de gravedad, ya que esto hace que el movimiento sea mucha más eficiente (“Biomedical analysis of body mass transfer during stair ascent and descent of healthy subjects”, 1993).

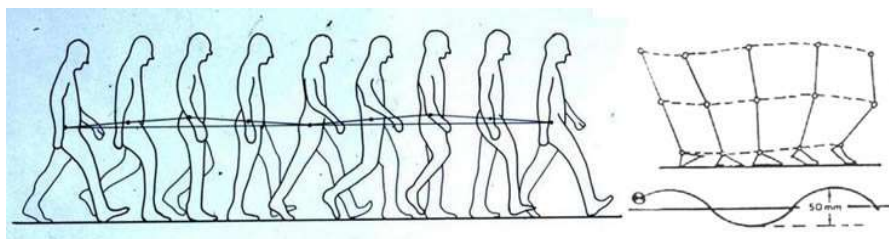


Figura 2.3: Trayectoria típica de desplazamiento del centro de gravedad de una persona durante el ciclo de marcha. Fuente: <https://m.exam-10.com/pravo/6637/index.html?page=4>

Por el contrario, cuando subimos escaleras lo que hacemos es elevarnos a nosotros mismos, y por lo tanto a nuestro centro de gravedad, lo que supone un gran gasto de energía extra en comparación a cuando caminamos, esta energía extra que gastamos se traduce en un aumento de la energía potencial de nuestro cuerpo. Este aumento en la energía potencial estará directamente relacionado con la altura del escalón. Además, cuando subimos escaleras necesitamos un rango de movimientos en las articulaciones inferiores (tobillo, rodilla y cadera) más amplio.

Al igual que en el ciclo de marcha, podemos dividir el ciclo de ascenso de escaleras en diferentes fases. Tomando un ciclo como el periodo entre el contacto en el escalón de uno de los pies, hasta el momento en el que el pie vuelve a entrar en contacto con el siguiente escalón, cada pie pasa por la fase de apoyo ($65\% \pm 4\%$ del ciclo total) y la fase de oscilación ($35\% \pm 4\%$ del ciclo total) (“Stair ascent and descent at different inclinations”, 2001).

Las subfases de la fase de apoyo incluyen; el contacto con el pie (0-2% del ciclo total); aceptación el peso (0-17% del ciclo total); empuje vertical (2-37% del ciclo total); soporte de una sola extremidad (17-48% del ciclo total); la continuación hacia delante (37-51% del ciclo total); y doble apoyo (48-65% del ciclo total). La duración total de la secuencia en el caso de este estudio resulta en un tiempo medio de 1.66 segundos con una desviación de ± 0.37 segundos.

La fase de oscilación está formada por dos subfases, una fase de elevación del pie (65-82% del ciclo total) y la fase de posicionamiento del pie en el escalón (82-100% del ciclo total).

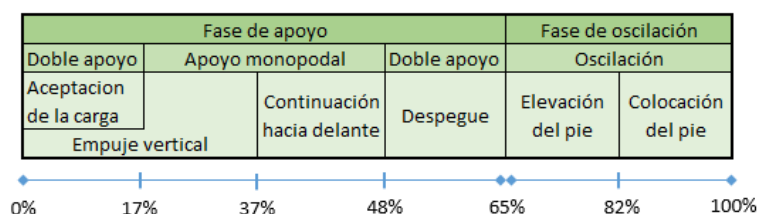


Tabla 2.1: Ciclo típico de una persona al subir escaleras. Fuente: Elaboración propia a partir de: https://www.researchgate.net/profile/James-Zachazewski/publication/15039814_Biomedical_analysis_of_body_mass_transfer_during_stair_ascent_and_descent_of_healthy_subjects/links/0912f511a9dcce178b000000/Biomedical-analysis-of-body-mass-transfer-during-stair-ascent-and-descent-of-healthy-subjects.pdf

Durante la fase de empuje vertical es donde se produce la mayor diferencia con la marcha humana. El centro de gravedad del cuerpo entero es desplazado verticalmente a lo largo de la subfase de empuje vertical en la fase de apoyo, en este caso, el desplazamiento producido será igual a la altura del peldaño de la escalera. Durante la subfase de continuación hacia adelante, no se produce ningún desplazamiento adicional del centro de gravedad en el eje vertical.

El centro de gravedad también sufre un desplazamiento lateral que describe una curva sinusoidal en el plano frontal. El desplazamiento máximo (4.4 ± 1.2 cm) ocurre en torno a la mitad de la fase de apoyo, hacia el 34% del ciclo total, durante la subfase de empuje vertical. Sigue una rápida inversión de dirección, y el peso se transfiere a la extremidad opuesta después de un golpe con el pie opuesto durante la mitad de doble soporte (59% del ciclo total) (“Stairclimbing Kinematics on Stairs of Differing Dimensions”, 1991).

2.3 Patrones de marcha en robots

La locomoción en robots bípedos no es una novedad tecnológica, se llevan desarrollando estrategias para la marcha estable desde 1970, cuando se inició la investigación del robot WABOT 1, desarrollado por la Universidad de Waseda. Pese a no ser una novedad, sí es un campo todavía de desarrollo y con mucho margen de mejora.

Este capítulo tiene como objetivo un estudio del arte de las técnicas de locomoción para robots humanoides, que nos ayudará, más adelante a diseñar las trayectorias articulares que debemos seguir para conseguir que el robot Nao suba escaleras.

2.3.1 Zero-Moment Point

No podemos empezar de otra forma este capítulo que no sea hablando del Zero Moment Point (ZMP), que es el método más extendido para planificación de patrones de marcha para robots bípedos.

Este método fue desarrollado por Miomir Vukobratovic alrededor de 1969 a pesar que el termino ZMP se acuñó unos cuantos años después. No fue hasta 19984 cuando este criterio pudo ser aplicado con éxito, en la Universidad de Waseda, en el robot WL-10RD (figura 2.4), el primero equilibrado dinámicamente (Miomir Vukobratovic, 2004).

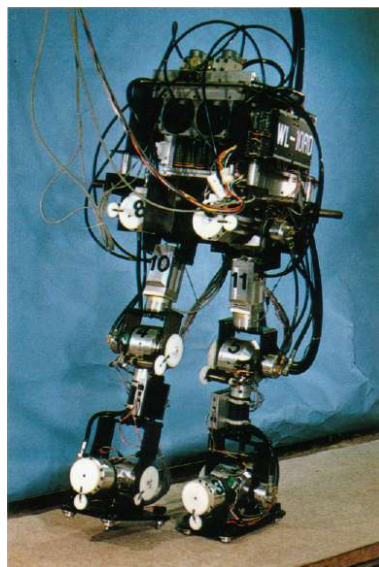


Figura 2.4: Robot WL-10RD, Fuente: <http://sashrobo.blogspot.com/2015/01/shasher.html>

El método Zero Moment Point se define como, el punto con respecto al cual la fuerza de reacción dinámica producida por el contacto del pie con el suelo no produce ningún momento de inercia en dirección horizontal, o lo que sería lo mismo, el punto donde la suma de la totalidad de las fuerzas horizontales, contabilizando la inercia y la gravedad sea igual a cero (Oort y Stramigioli, 1996).

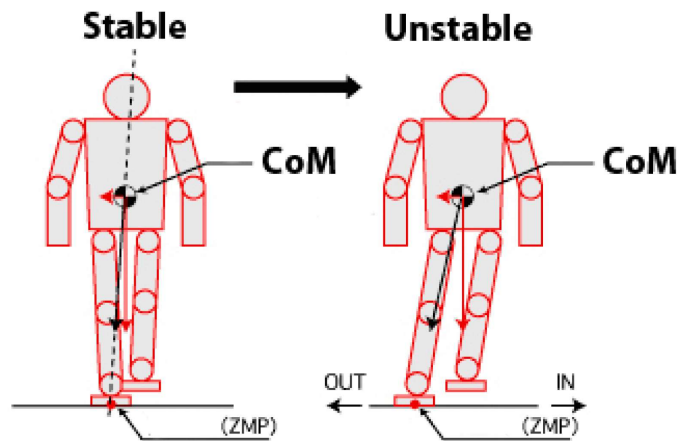


Figura 2.5: Representación gráfica del concepto ZMP, Fuente: <https://www.mdpi.com/2076-3417/9/3/502/htm>

La estabilidad del sistema se garantiza si el ZMP está dentro de una región denominada polígono de soporte (SP) que es la superficie formada por una envolvente convexa de todos los puntos de apoyo con el suelo. En el caso de un robot sus plantas de los pies. Como hemos visto anteriormente en la marcha bípeda podemos distinguir dos fases de apoyo de los pies, una fase en la que ambos pies están apoyados y otra fase donde uno de los dos pies está en el aire. En estos casos el polígono de soporte quedaría como se puede ver en la figura 2.6:

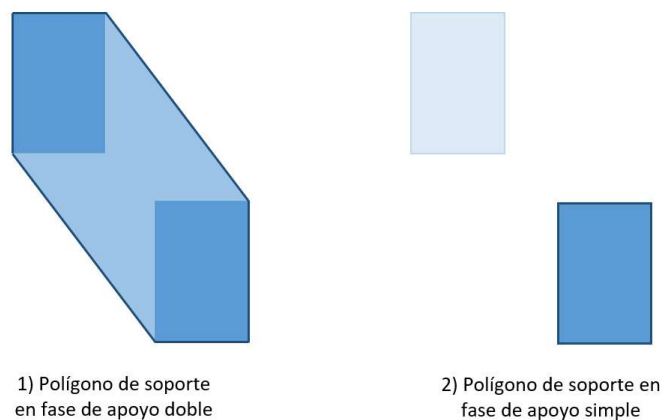


Figura 2.6: Polígono de soporte para robot bípedo. Fuente: Elaboración propia

Para posicionar el robot en el espacio, se utilizará el base-frame-origin, y tres planos perpendiculares entre sí, esto lo podemos visualizar en la figura 2.7, donde podemos ver un humanoide en “posición anatómica” (robot de pie, mirando hacia delante y los brazos estirados a ambos lados del cuerpo, las palmas mirando hacia delante y los pies juntos). El movimiento puede describirse de forma relativa a los tres planos perpendiculares:

Frontal (plano YZ), sagital (plano XZ) y transversal (plano XY).

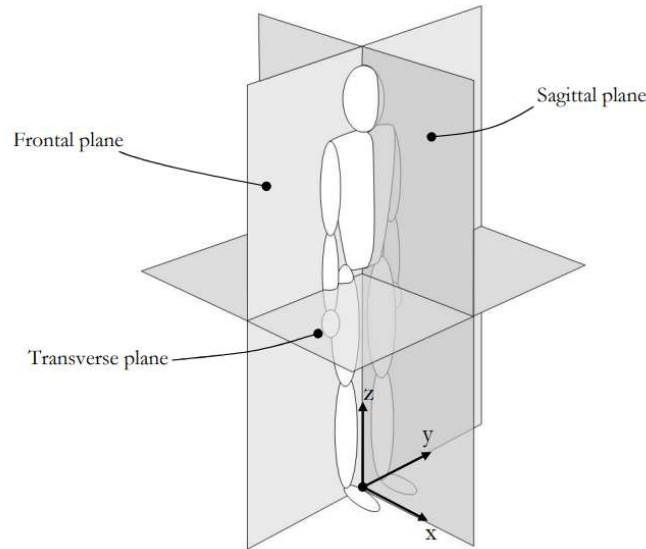


Figura 2.7: Humanoide en “posición anatómica” incluyendo representación de planos y origen, Fuente: https://www.techunited.nl/media/files/humanoid/MaartenDekker_OPEN2009_Zero_Moment_Point_Method_for_Stable_Biped_Walking.pdf

Si consideramos un pie rígido con una suela plana que está completamente en contacto y apoyado por el piso, como se muestra en la figura 2.8. Para simplificar, el robot se reemplaza por una fuerza F_A y el momento M_A que actúa sobre un punto A en el suelo. La aceleración gravitacional es g , actuando en la dirección Z negativa.

El objetivo del método ZMP es encontrar el punto “P” a lo largo del plano del suelo donde F_{PZ} compense las componentes horizontales de M_A , (M_{AX} y M_{AY}). Con el fin de mantener el humanoide en equilibrio: en el punto P actúan la fuerza de reacción $F_P = (F_{PX}, F_{PY}, F_{PZ})$ y el momento $M_Z = (M_{PX}, M_{PY}, M_{PZ})$. Suponiendo que no se produce deslizamiento entre el suelo y el pie del robot, la fuerza de reacción horizontal (F_{PX}, F_{PY}) se verá compensada con las componentes horizontales de la fuerza F_A , (F_{AX}, F_{AY}). Lo mismo debe ocurrir con la componente vertical del momento M_{PZ} y M_{AZ} , la componente vertical del momento de reacción, que es M_{PZ} , debe estar equilibrando la componente vertical del momento M_A y el momento inducido por la fuerza F_A .

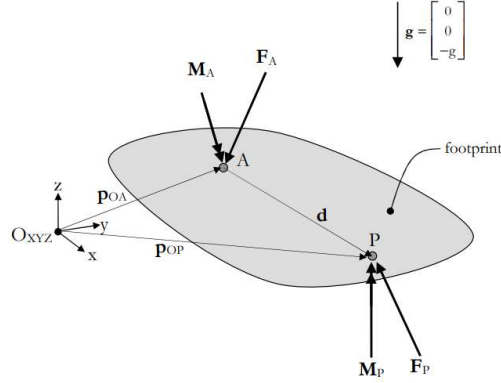


Figura 2.8: Fuerzas y momentos actuando en un pie rígido en contacto con el suelo. Fuente: https://www.techunited.nl/media/files/humanoid/MaartenDekker_OPEN2009_Zero_Moment_Point_Method_for_Stable_Biped_Walking.pdf

Para lograr un patrón de marcha dinámica estable el ZMP debe estar en todo momento dentro del polígono de soporte. Si queremos calcular la posición del ZMP a partir de una trayectoria, debemos imponer como condición las siguientes ecuaciones:

$$F_P + F_A = 0 \quad (2.1)$$

$$p_{OP} \times F_P + M_A + M_{PZ} + p_{OA} \times F_A = 0 \quad (2.2)$$

Los términos p_{OP} y p_{AP} son vectores desde el base-frame-origin O_{XYZ} a los puntos P y A respectivamente. Si el base-frame-origin se posiciona en el plano XY la ecuación 2.2 quedaría como:

$$(p_{OP} \times F_P)_{XY} + M_{AXY} + (p_{OA} \times F_A)_{XY} = 0 \quad (2.3)$$

En resumen, siempre que la reacción del suelo debido al contacto del pie con él pueda reducirse a una fuerza F_P y a una componente vertical M_{PZ} , dicho punto se considerará el ZMP. En caso de que el punto P se encuentre fuera del polígono de soporte la fuerza F_P actuará en el extremo del pie, con lo que no se compensarán ni F_A ni M_A , con lo cual el robot caerá al suelo.

Mientras la ecuación 2.3 se cumpla el robot no perderá el equilibrio.

Si la dinámica del robot cambia de forma que el punto P alcanza el borde del polígono de soporte el sistema se convierte en un sistema marginalmente estable, por lo tanto, cualquier pequeña perturbación podría hacer que definitivamente, el punto P saliera del polígono de soporte y el sistema se volviese inestable, así que debemos considerar que el punto P debe estar dentro de la zona del polígono de soporte sin tener en cuenta el perímetro o el borde, del polígono de soporte.

2.3.1.1 Centro de Presión (CoP) y su relación con el ZMP

El centro de presión (CoP), es el punto dentro del polígono de soporte del humanoide, donde la suma de todas las fuerzas de contacto actúa, causando una fuerza, pero no un momento. Esto sucede cuando no tenemos deslizamiento entre la suela del pie del robot y el suelo, por lo tanto, las fuerzas de fricción se anulan. La fuerza normal resultante de este campo de presión con un número "n" de puntos de contacto se expresa como:

$$F_{RN} = \sum_{i=1}^n F_{Ni} \quad (2.4)$$

Y la posición del CoP respecto del base-frame-origin, expresado como p_{CoP} , se calcula como:

$$P_{CoP} = (\sum_{i=1}^n p_{F_{Ni}} * F_{Ni}) / F_{RN} \quad (2.5)$$

Si el ZMP coincide con el CoP el sistema es dinámicamente estable. En caso contrario, ambos elementos están desalineados. Cuando el sistema es dinámicamente estable se puede calcular el ZMP mediante la computación del CoP, por ejemplo, con sensores de fuerza en la suela del pie.

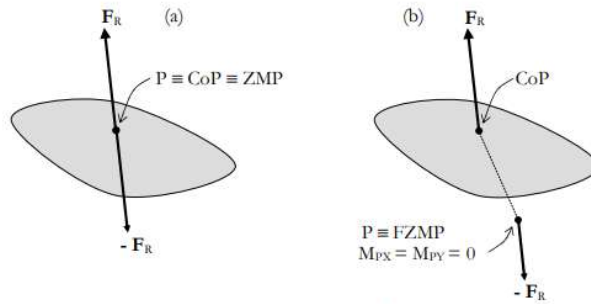


Figure 3.2: Relationship between ZMP, FZMP and CoP:
(a) Dynamically stable. (b) Dynamically unstable.

Figura 2.9: Relación entre el ZMP y el CoP: (a) Dinámicamente estable. (b) Dinámicamente inestable. Fuente: https://www.techunited.nl/media/files/humanoid/MaartenDekker_OPEN2009_Zero_Moment_Point_Method_for_Stable_Biped_Walking.pdf

2.3.1.2 Cálculo del ZMP por cinemática directa

Para calcular el punto P, hay que realizar las siguientes consideraciones:

1. El humanoide está formado por "n" eslabones rígidos.
2. Toda la información cinemática, como la posición del centro de masas, la orientación de los eslabones, velocidades etc. son conocidos y calculados mediante cinemática directa.
3. El suelo es rígido y estable.
4. El pie del robot no desliza en su contacto con el suelo.
5. Todas las articulaciones del robot están actuadas.

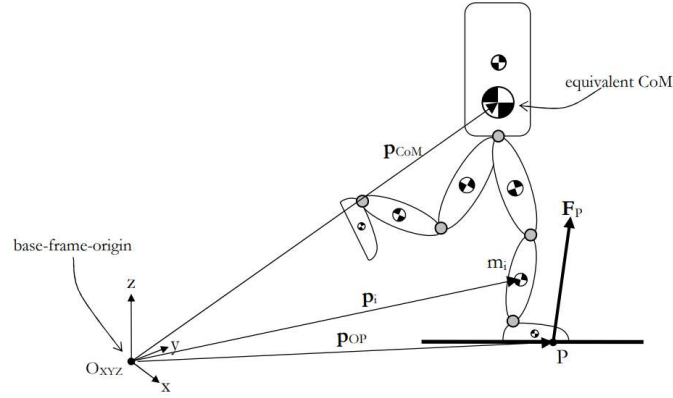


Figura 2.10: Modelo esquemático del robot bípedo y el punto P. Fuente: https://www.techunited.nl/media/files/humanoid/MaartenDekker_OPEN2009_Zero_Moment_Point_Method_for_Stable_Biped_Walking.pdf

Bajo estas condiciones, primero, deberemos calcular la masa total del humanoide (m_{tot}) y la distancia desde el base-frame-origin al centro de masas (p_{CoM}).

$$m_{tot} = \sum_{i=1}^n F_{mi} \quad (2.6)$$

Con esta información, el momento lineal total P y el momento angular total H respecto del base-frame-origin puede expresarse como:

$$P = \sum_{i=1}^n m_i * \dot{p}_i \quad (2.7)$$

$$H = \sum_{i=1}^n p_i \times m_i * \dot{p}_i + I_i * w_i \quad (2.8)$$

En la ecuación 2.8, I_i y w_i son respectivamente el tensor de inercia y la velocidad angular de cada eslabón "i" respecto al origen. El tensor de inercia se calcula como:

$$I_i = R_i * I_i * R_i^T \quad (2.9)$$

Se calculan las derivadas del momento lineal y del momento angular, P y H respectivamente:

$$\dot{P} = \sum_{i=1}^n m_i * \dot{p}_i \quad (2.10)$$

$$\dot{H} = \sum_{i=1}^n (\dot{p}_i \times (m_i * \dot{p}_i) \times ((m_i * \ddot{p}_i)) + I_i * \dot{w}_i + w_i \times (I_i * w_i)) \quad (2.11)$$

Donde $p_i \times (m_i p_i) = 0$ porque p_i y $p_i * m_i$ son paralelos. Una vez obtenidos P' y H' se sustituyen en las ecuaciones de equilibrio dinámico (2.1 y 2.2) descritas anteriormente:

$$F_P = -F_A = \dot{P} - m_{tot} * g \quad (2.12)$$

$$M_O = -\dot{H} - p \times m_{tot} * g \quad (2.13)$$

Como hemos visto anteriormente, F_P y M_O son la fuerza y el momento externo que el suelo ejerce sobre el robot con respecto al base-frame-origin. Y F_A es la fuerza que el robot ejerce contra el suelo. La fuerza F_P actúa sobre el punto P, así que M_O se define como:

$$M_O = p_{OP} \times F_p + M_p \quad (2.14)$$

Sustituyendo la ecuación 2.12 en 2.13:

$$M_p = \dot{H} - p_{CoM} \times m_{tot}g + (\dot{P} - m_{tot}g) \times p_{OP} \quad (2.15)$$

La distancia del base-frame-origin al ZMP es igual a p_{OP} ($p_{ZMP} = p_{OP}$), por lo tanto:

$$X_{ZMP} = (m_{tot} * g_z * p_{CoMx} + Z_{ZMP} * \dot{P}_x - \dot{H}_y) / m_{tot} * g_z + \dot{P}_z \quad (2.16)$$

$$Y_{ZMP} = (m_{tot} * g_z * p_{CoMy} + Z_{ZMP} * \dot{P}_y - \dot{H}_x) / m_{tot} * g_z + \dot{P}_z \quad (2.17)$$

2.3.1.3 Modelo Cart-Table

Un modelo simplificado para calcular el ZMP es el llamado modelo Cart-Table. Como se ilustra en la figura 2.11, se representa el modelo del robot bípedo como un carro desplazándose sobre una mesa de masa despreciable. El carro tiene una masa m y su posición (x, z) se corresponde con el centro de masas general del robot humanoide. Y la mesa tiene el mismo polígono de soporte que el robot.

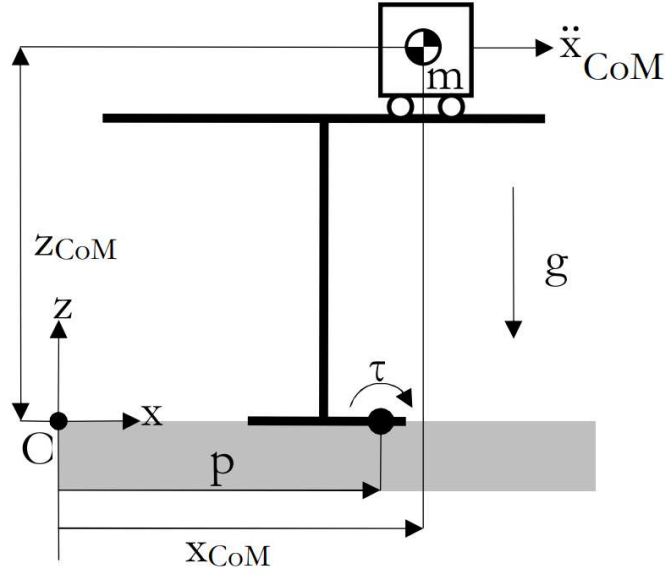


Figura 2.11: Modelo Cart-Table. Fuente: https://www.techunited.nl/media/files/humanoid/MaartenDekker_OPEN2009_Zero_Moment_Point_Method_for_Stable_Biped_Walking.pdf

En este caso el momento τ , ejercido en el punto P se describe como:

$$\tau = -mg(x_{CoM}-p) + m * \ddot{x}_{CoM} * z_{CoM} \quad (2.18)$$

Ahora usando la definición de ZMP, que nos dice que en el punto P τ tiene que anularse, y que $X_{ZMP} = p$, la posición del ZMP queda definida como:

$$X_{ZMP} = p = x_{CoM} - (\ddot{x}_{CoM}/g) * z_{CoM} \quad (2.19)$$

$$Y_{ZMP} = p = y_{CoM} - (\ddot{y}_{CoM}/g) * z_{CoM} \quad (2.20)$$

(Dekker, 2009)

2.3.1.4 Limitaciones del modelo ZMP

La marcha humana es muy compleja, implica el movimiento coordinado de muchas articulaciones y músculos, además que los elementos que componen el cuerpo humano muchas veces son difícilmente reproducibles en un robot, que generalmente se componen de elementos rígidos. Por esta razón una de las limitaciones del ZMP es el hecho de que no se consigue un movimiento de marcha que se asemeje a la humana, es un tipo de marcha muchos menos fluida y natural.

2.3.2 Estrategias de Marcha Alternativas al ZMP

2.3.2.1 Marcha Pasiva

La marcha pasiva es una aproximación muy diferente a los patrones de marcha vistos hasta ahora, esta aproximación de dinámica pasiva, no extrae la energía para su funcionamiento de ningún tipo de fuente convencional, ya sea combustible fósil o una batería, sino que aprovecha su energía potencial, esta es la principal limitación de este tipo de mecanismos, sólo pueden realizar patrones de marcha en superficies con una ligera inclinación negativa, es decir, sólo pueden ir cuesta abajo. Este principio fue desarrollado por Tad McGeer durante la década de 1980.



Figura 2.12: Dynamite, “Passive Dynamic Walker” con rodillas de McGeer. Fuente: <https://intranet.ceautomatica.es/old/actividades/jornadas/XXIX/pdf/240.pdf>

McGeer demostró que un cuerpo humano puede caminar por una pendiente sin requerir músculos o motores. A diferencia de los robots tradicionales, que gastan energía mediante el uso de motores para controlar cada movimiento, las máquinas pasivo-dinámicas de McGeer se basan solo en la gravedad y el balanceo natural de sus extremidades para avanzar cuesta abajo.

Los prototipos desarrollados por McGeer tenían unas características que les hacían muy diferentes a los robots humanoides tradicionales. A pesar de la ausencia de control, puesto que no contaban con ningún tipo de actuador eran capaces de caminar de una forma más parecida a la de un humano, en comparación con los robots humanoides “actuados” que se habían presentado hasta la fecha.

Cuando una de las piernas comienza su fase de apoyo, ésta se comporta como un péndulo invertido, al final de la fase de oscilación la pierna en balanceo golpea el suelo, disipándose energía. Es la energía potencial que aporta el descenso de la rampa quien se encarga de compensar esta pérdida, lo que hace posible que el modelo camine siempre con la distribución de masas y longitudes adecuada, se consigue un ciclo estable que se repite indefinidamente.

Inicialmente el mecanismo no tenía rodillas, pero posteriormente, en 1990 McGeer las implementó. Resolviendo el problema de los pies que chocan con el suelo cuando la pierna se balancea hacia adelante y también ofrecen más estabilidad en algunos entornos. Otra particularidad importante, es que tenían los pies curvos, lo que impedía que fueran controlables, ignorando la necesidad de estabilidad local. El movimiento lateral desaparece al contar con cuatro piernas (“Síntesis de la marcha robótica mediante la aproximación del ciclo límite: El Robot Bípedo Esbirro”, Fecha de la última visita: 08/06/21).

2.3.2.2 Passive Dynamic Walking

A raíz de los prometedores resultados obtenidos por las máquinas pasivas de McGeer, comenzó a explorarse la posibilidad de sustituir la energía que aportaba la pendiente por algún tipo de actuador. Como se verá a continuación, los primeros prototipos incorporaban tan sólo actuación en una de sus articulaciones, típicamente en la cadera o el tobillo. Recientemente se ha definido un marco formal que contiene a estos robots basados en “Pasive Dynamic Walkers”, bautizándolos Robots de Ciclo Límite (“Limit Cycle Walkers”), debido a que se basan en la generación de marcha (nominalmente) cíclica (*Efficient Bipedal Robots Based on Passive-Dynamic Walkers*, 2005).

A modo de ejemplo veremos tres robots que usan este tipo de estrategia de locomoción:

- El robot bípedo de Cornell, este robot está impulsado por un motor eléctrico con muelles que actúa sobre el movimiento del tobillo. Tiene 5 grados de libertad (cadera, los dos tobillos y las rodillas) y los brazos están unidos a la pierna opuesta.
- El robot bípedo de la Universidad Técnica de Delph, llamado “Denise” presenta unas características morfológicas parecidas, pero en este caso la articulación actuada es la de la cadera y es de tipo neumático.
- Por último, tenemos el robot “Toddler” diseñado por el Massachusetts Institute of Technology (MIT), posee 6 grados de libertad, con dos servomotores actuando cada uno de los tobillos y dos articulaciones en la cadera pasivas, al igual que en el bípedo de Cornell, los brazos están unidos a la cadera opuesta. Un rasgo peculiar de este robot es que el control de la actuación mecánica se determina usando “reinforcement learning”, una técnica de inteligencia artificial basada en aprender de la experiencia, esta técnica se basa en un sistema de recompensas que premia al robot cuando consigue llegar a los objetivos establecidos, para ello el robot va haciendo probando acciones aleatorias hasta que consigue el patrón de acciones que maximiza la recompensa.

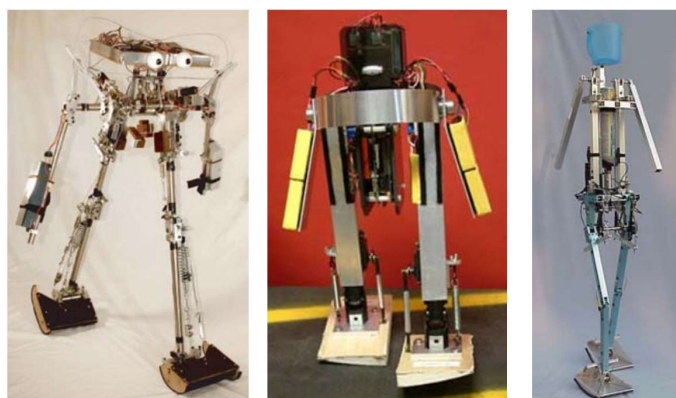


Figura 2.13: De derecha a izquierda: “Toddler” del MIT, Cornell Efficient Biped, “Denise” de la TU Delft, Fuente: <https://intranet.ceautomatica.es/old/actividades/jornadas/XXIX/pdf/240.pdf>

Los robots con este tipo de estrategia de locomoción realizan un uso de la energía muy eficiente, el bípedo de Cornell, que destaca especialmente en este ámbito, consume 11 Vatios.

Para poder hacer una comparativa realista entre humanos y robots de diferentes tamaños se usa el coste específico de energía adimensional (C_t), cuya fórmula matemática surge de dividir la energía requerida entre el peso del robot multiplicado por la distancia cubierta:

$$C_t = (Energy_{used}) / (weight \times distance_{travelled}) \quad (2.21)$$

Con el fin de aislar la efectividad del diseño mecánico y el controlador, de la eficiencia de la actuación, distinguiremos entre el coste energético específico de transporte, C_{et} , y el coste mecánico específico de transporte, C_{mt} .

Donde el C_{et} usa la energía total consumida por el sistema, el C_{mt} solo considera el trabajo mecánico del actuador. El bípedo de Cornell caminando a 0.4 m/s tiene un C_{et} de 0.2 y un C_{mt} de 0.055. Unos valores muy próximos a los del ser humano. Los robots de Delft y del MIT no están diseñados centrándose especialmente en conseguir una buena eficiencia, aun así, al ser robots basados en la estrategia de locomoción passive-dynamic walking también obtienen muy buenos resultados, con un C_{mt} de 0.08 y 0.02, respectivamente. Por el contrario, otros robots que se basan en algoritmo ZMP para caminar son varios ordenes de magnitud menos eficientes. Por un lado el robot NAO, usando el patrón de marcha predefinido por el fabricante, tiene un C_{mt} de 5.8, mientras que el ASIMO de Honda tiene un C_{mt} de 1.5. Ambos muy alejados de los valores de eficiencia que alcanzan los “passive-dynamic walkers” (*Efficient Bipedal Robots Based on Passive-Dynamic Walkers*, 2005).

Por último, cabe destacar la sencillez en el control de este tipo de robots, los algoritmos de control de marcha suelen ser complejos, requiriendo de una alta capacidad de computación en tiempo real. Por el contrario, el bípedo de Delft y de Cornell caminan haciendo uso de algoritmos muy primitivos, sus sensores detectan cuando hay contacto con el suelo, y sus únicos comandos para el motor son de On/Off una única vez por paso.

3 Objetivos

Una vez conocida la historia de la robótica humanoide hasta la actualidad y habiendo presentado el marco teórico y el estudio de las diferentes técnicas implementadas en el arte de la locomoción de robots humanoides bípedos. Se procede a establecer los objetivos a alcanzar con este proyecto.

- Síntesis de los conceptos teóricos más importantes de los que se nutren las estrategias de locomoción robótica que han sido estudiadas en esta memoria, para que, puedan ser puestas en práctica en el diseño un patrón de movimientos, basado en trayectorias articulares de las diferentes articulaciones que componen el tren inferior del robot NAO, que le permita a éste poder subir escalones sin perder el equilibrio.
- Familiarizarse con el robot NAO y conocer las posibilidades que brinda su entorno de desarrollo para así poder llegar a resolver el desafío de la forma más eficiente posible.
- Dotar al programa principal de desarrollo del robot NAO, Choregraphe, una nueva funcionalidad, en forma de bloque, que permita al robot subir escaleras, tanto usando su pierna derecha como su pierna izquierda para subir el escalón.
- Representar y analizar las trayectorias temporales de las articulaciones y del resto de datos obtenidos durante las pruebas.

4 Metodología

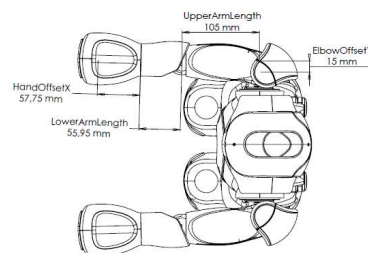
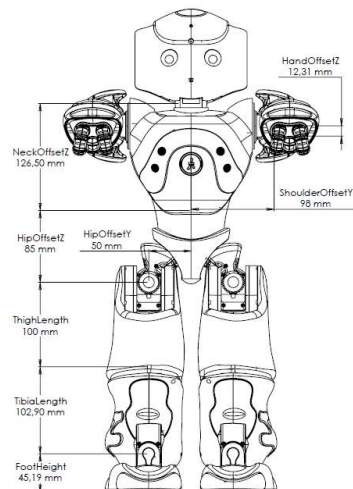
4.1 Entorno de trabajo

4.1.1 Robot NAO

El robot humanoide NAO es el robot con el que se va a trabajar en este proyecto. NAO fue el primer robot creado por la compañía francesa Aldebaran Robotics, subsidiaria del grupo Softbank Robotics. La primera versión se lanzó en el año 2006, pero el robot no ha dejado de evolucionar, ahora mismo se encuentra en su versión número 6 (NAO6), que es con la que vamos a trabajar. Esta nueva versión integra una nueva CPU que mejora su potencia de cálculo y su rendimiento en general (*Documentación NAO 2.8*, Fecha de la última visita: 03/07/21).

4.1.1.1 Características técnicas

Nao es un robot humanoide de 574 mm de alto, pesa 4.3 kilogramos, su carcasa está fabricada en diferentes plásticos duros, alternando acabados mate y brillante.



(a) Vista frontal. Fuente: http://doc.aldebaran.com/2-8/family/nao_technical/links_naov6.html

(b) Vista Superior. Fuente: http://doc.aldebaran.com/2-8/family/nao_technical/links_naov6.html

Figura 4.1: Dimensiones del Robot NAO

Su placa base está compuesta por un procesador Atom E3845 de cuatro núcleos a 1.91 GHz y 4 GB de memoria RAM. En cuanto al sistema operativo que gobierna el sistema, estamos ante un sistema operativo propio NAOqi OS, que es una distribución GNU/Linux basado en Yocto. Su batería tiene una capacidad de 62.5 Wh que le confiere aproximadamente una hora de funcionamiento sin estar conectado por cable.

En la parte posterior de la cabeza encontramos los puertos que nos permiten conectarnos al robot, disponemos de una conexión RJ45 y un puerto USB, además nos podemos comunicar con él mediante Wi-fi IEEE 802.11. En los laterales de su cabeza tenemos dos altavoces que le permiten comunicarse con el exterior. Además, está equipado con 4 micrófonos unidireccionales que le permiten comunicarse con el exterior. En la frente del robot y en su boca encontramos dos cámaras idénticas de 5 megapíxeles que proporcionan una resolución de 2560x1920 píxeles a un fotograma por segundo o 640x480 píxeles a 30 fotogramas por segundo.

El robot cuenta con un total de 25 grados de libertad que podemos segmentar en diferentes cadenas cinemáticas para su mejor visualización:

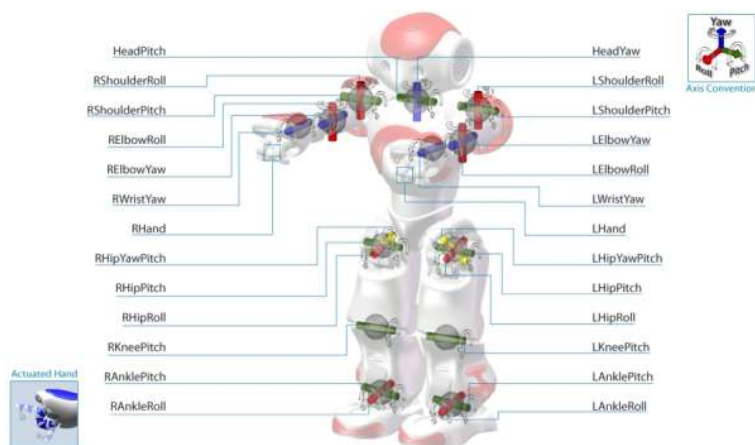


Figura 4.2: Vista de las articulaciones del robot NAO. Fuente: http://doc.aldebaran.com/2-8/family/nao_technical/lola/actuator_sensor_names.html

Las articulaciones están actuadas por diferentes tipos de motores eléctricos, acompañados de dos juegos de reducciones para adecuar la velocidad del movimiento a las necesidades de la articulación.

Head (Cabeza)	HeadYaw (Rotación de cabeza)
	HeadPitch (Movimiento de cabeceo)
R/L Arm	R/LShoulderPitch (Flexión de hombro)
	R/LShoulderRoll (Abducción de hombro)
	R/LElbowYaw (Rotación de codo)
	R/LElbowRoll (Flexión de codo)
	R/LWristYaw (Rotación muñeca)
	R/Lhand (Abrir/Cerrar mano)
R/L Leg	R/LHipPitch (Flexión cadera)
	R/LHipRoll (Abducción cadera)
	R/LAnklePitch (Flexión tobillo)
	R/LAnkleRoll (Abducción tobillo)
	R/LKneePitch (Flexión rodilla)
Torso	HipYawPitch

Tabla 4.1: Tabla cadenas cinemáticas y articulaciones del NAO. Fuente: Elaboración propia a partir de: http://doc.aldebaran.com/2-8/family/nao_technical/bodyparts_naov6.html

	Tipo de Motor 1	Tipo de Motor 2	Tipo de Motor 3	Tipo de Motor 4	Tipo de Motor 5
Modelo	22NT Z13	17N	16GT	DCX 16S	22NT Z20
Velocidad sin carga	8 700 rpm $\pm 10\%$	8 400 rpm $\pm 12\%$	10 700 rpm $\pm 10\%$	11 400 rpm $\pm 10\%$	8 700 rpm $\pm 10\%$
Par máximo	65 mNm $\pm 8\%$	9.4 mNm $\pm 8\%$	14.3 mNm $\pm 8\%$	22.4 mNm $\pm 10\%$	65 mNm $\pm 8\%$
Par nominal	17.8 mNm max	4.9 mNm max	6.2 mNm max	2.6 mNm max	17.8 mNm max
	Ratio de reducción de la velocidad A				
	201,3	50,61	150,27	150,27	130,85
Articulaciones	HipYawPitch, R/LHipRoll, R/LAnkleRoll	R/LWristYaw	HeadYaw, R/LElbowYaw	R/LShoulderPitch	R/LHipPitch, R/LKneePitch, R/LAnklePitch
	Ratio de reducción de la velocidad B				
		36,24	173,22		
		R/LHand	HeadPitch, R/LShoulderRoll, R/LElbowRoll		

Tabla 4.2: Table de los diferentes tipos de motores delNAO. Fuente: Elaboración propia a partir de http://doc.aldebaran.com/2-8/family/nao_technical/motors_naov6.html

4.1.2 Software de desarrollo

4.1.2.1 NAOqi OS

NAOqi OS es el sistema operativo del robot, que es una distribución embebida GNU/Linux basada en Yocto, especialmente desarrollada para adecuarse a las necesidades del robot. Este software nos facilita y corre las librerías y los programas requeridos por NAOqi, el software principal del robot. El robot no puede realizar ninguna acción si NAOqi no está funcionando (*Documentación NAOqi 2.8*, Fecha de la última visita: 03/07/21).

El *Framework* de NAOqi da acceso a diferentes módulos que te permiten acceder a las diferentes funcionalidades del robot y gestionar su comportamiento, a través de los módulos de programas y métodos gestionados por este software intermedio. Estos módulos dan acceso a las funcionalidades básicas del robot, como a las cámaras, a los micrófonos o al control de posición de las articulaciones. Pero también hay módulos con funcionalidades más avanzadas, como el módulo de detección de caras humanas, que procesa las imágenes y da información de los rasgos más relevantes de la fisionomía de las caras que detecte, otra función de alto nivel que implementa es un algoritmo de locomoción basado en ZMP.

Cuando se construye una aplicación en este entorno, la aplicación se conecta a estos módulos para recibir y enviarles información a estos módulos. También es posible implementar nuevos módulos con nuevas funcionalidades usando el kit de desarrollo de software SDK. NAOqi se puede utilizar en nuestro ordenador, descargando el SDK correspondiente al lenguaje que se vaya a utilizar, C++ y Python son los dos lenguajes que tienen soporte.

4.1.2.2 Choreographe

Además, SoftBank Robotics nos proporciona una aplicación de escritorio, Choreographe, en la cual es posible programar el comportamiento del robot de forma gráfica, con un lenguaje de programación por bloques, haciendo uso de bloques predefinidos con funcionalidades complejas y con ciertos parámetros a ajustar por el usuario. Estos bloques se organizan en un flujo de trabajo, cada bloque se une al flujo de trabajo mediante una entrada que activa la funcionalidad del bloque, y una o varias salidas, que generan un pulso de activación para los bloques posteriores si los requerimientos para su activación, definidos por el usuario, se han cumplido.

Los bloques se encuentran dentro de la pestaña “Box libraries” categorizados por carpetas según sus funcionalidades. Cuando nuestro programa es complejo y es necesario usar muchos bloques para su configuración es interesante usar los Flow diagram box, que permiten encapsular bloques dentro de ellas, creando así un nuevo subespacio de trabajo.

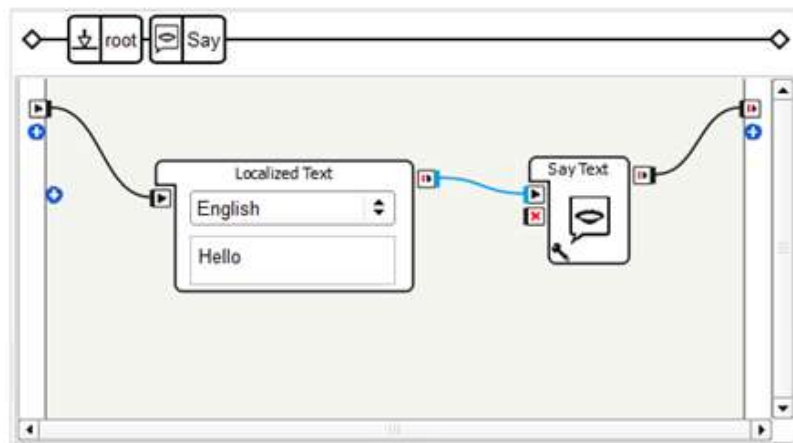


Figura 4.3: Workspace de Choregraphe con bloques de funciones. Fuente: http://doc.aldebaran.com/2-8/software/choregraphe/objects/flow_diagram_box.html

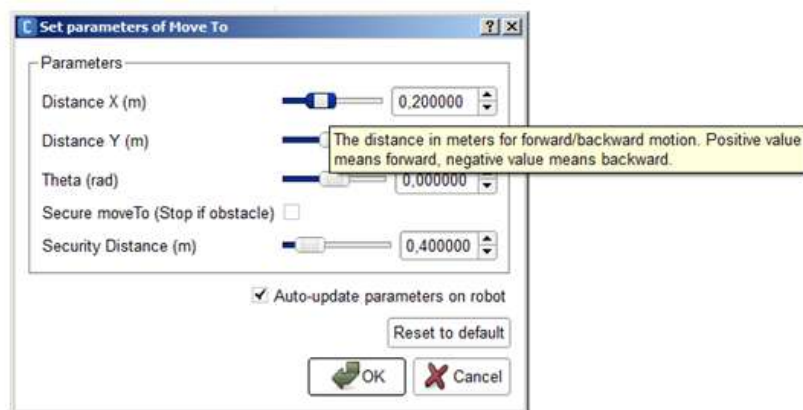


Figura 4.4: Vista del editor de parámetros para la función "Move To". Fuente: Elaboración propia

Choregraphe pone a disposición del usuario una extensa biblioteca de bloques, que permiten dotar al robot NAO de comportamientos muy complejos, pero como es lógico, al contar con un número limitado de ellas siempre es posible que surjan necesidades que no estén contempladas en los bloques predefinidos. Para solucionar esto, en Choregraphe existe la posibilidad de crear nuestros propios bloques haciendo uso del lenguaje de programación Python. Cuando creamos un bloque de Python nos da la opción de definir el nombre del bloque y agregar una descripción de su funcionalidad, además de poder definir los parámetros en entrada/salida del bloque.

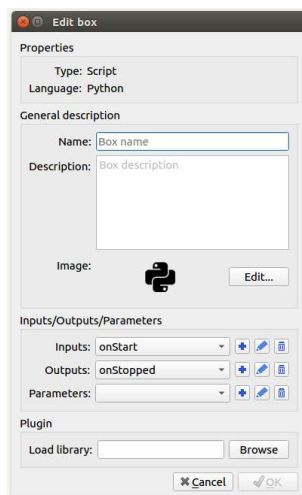


Figura 4.5: Vista de la pantalla de edición de un bloque Python. Fuente: Elaboración propia

Cuando se crea un bloque de programación Python éste nos aparece con unas líneas de código predefinidas, en estas líneas se genera una clase (todos los bloques tienen su propia clase, ya sean creados por el propio usuario o predefinidos), luego tenemos diferentes “Event handlers” o controladores de eventos, que generan respuestas a los diferentes eventos que pueden ocurrir. La parte de la programación más importante se hace en la función “OnStart”, donde se implementa la funcionalidad que va a tener el bloque Python, la función onStop, detiene la funcionalidad que se está llevando a cabo en el bloque y activa el “output” del bloque, señal que indica que la funcionalidad del bloque ha terminado:



Figura 4.6: Código predeterminado dentro de un bloque Python. Fuente: Elaboración propia

4.1.2.3 Monitor

Monitor es un programa incluido en la suite de Choregraphe, su funcionalidad principal es la de brindar información acerca del comportamiento de nuestro robot. La principal funcionalidad utilizada en este proyecto se encuentra dentro del módulo "Memory viewer" que es el complemento encargado de darnos información acerca de los datos que contiene el módulo ALMemory (que es el módulo donde se almacenan los datos de los motores y los sensores) de un NAOqi determinado. También se puede trazar la evolución de los datos en forma de números y gráficas. Esto es muy útil para diagnosticar comportamientos a través de los datos internos que utilizan.

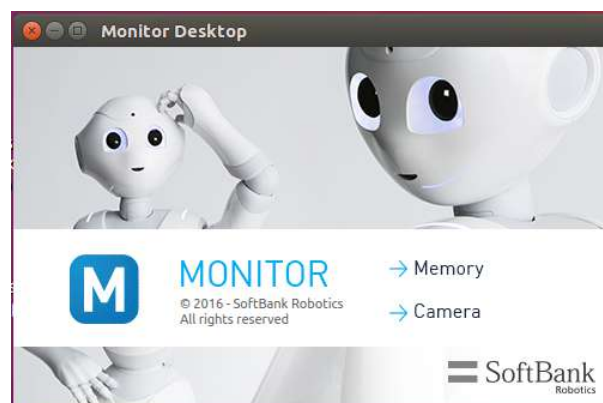


Figura 4.7: Vista principal de programa Monitor. Fuente: Elaboración propia

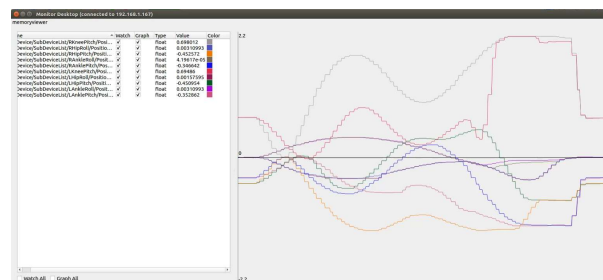


Figura 4.8: Gráfica del movimiento angular de las articulaciones del robot al subir un peldaño de la escalera. Fuente: Elaboración propia

4.2 Desarrollo

Cuando se plantea el diseño de un robot se parte de un lienzo en blanco, no hay ningún tipo de morfología obligatoria, de la que haya que dotar al robot de forma imperativa. Si se diseña un robot industrial, que va a trabajar dentro de su celda, se tenderá a un diseño que satisfaga los requerimientos de la tarea a realizar, y su morfología será la adecuada para realizar la tarea de una forma eficiente y económica.

Los robots humanoides generalmente no son robots enfocados a realizar tareas industriales, son robots con apariencia y morfología humana, y se les dota de estas características porque su principal cometido es interactuar con seres humanos. Por ello es vital que los robots puedan adaptarse al entorno en el que normalmente se mueven los humanos, y deben de poder desplazarse por este entorno sin limitaciones. Debido a esto, el poder subir escaleras, un elemento muy presente en cualquier edificio o entorno construido por los humanos, se convierte en una habilidad muy importante a la hora de moverse.

Esto confiere una ventaja importante a los robots bípedos de apariencia humana frente a otros robots de apariencia humana con ruedas, a los que se les hace muy difícil o imposible dependiendo del diseño de las ruedas subir un escalón (Ver ejemplo de este caso con el robot Pepper en la figura 4.9).

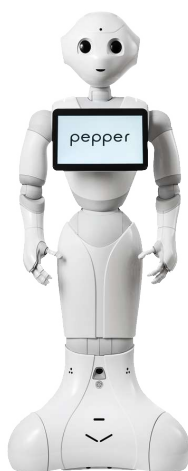


Figura 4.9: Robot humanoide Pepper. Fuente: <https://www.softbankrobotics.com/emea/es/pepper>

4.3 Estrategia propuesta para subir escaleras

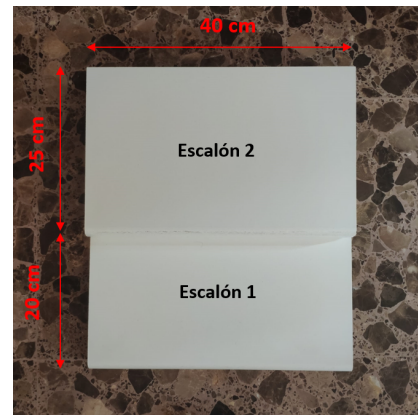
Una vez estudiadas las diferentes estrategias que siguen los robots para conseguir una marcha estable, se realiza una estrategia que permita al robot subir las escaleras de forma exitosa.

Se trabajará con los conceptos desarrollados en el apartado dedicado al ZMP (Zero-moment point) diseñando una trayectoria de forma empírica, trabajando directamente con el Robot NAO, para que éste sea capaz de subir escalones de 2.8 centímetros de altura, que, dado su estatura, equivaldría a un escalón de unos 9 centímetros para una persona de 180 centímetros de estatura.

En cuanto a las dimensiones físicas de los escalones utilizados en esta prueba, se trata de dos escalones, ambos con la altura ya mencionada de 2.8 centímetros de altura, y con un ancho de 40 centímetros, el primer escalón tiene una profundidad de 20 centímetros (figura 4.10b). Como el ancho de paso del movimiento que vamos a realizar es de en torno a 20 centímetros el robot no tendrá que realizar ninguna maniobra de aproximación hacia el siguiente escalón, puesto que este queda a la distancia justa para que el robot realice su segundo movimiento de ascensión, el segundo escalón tiene una profundidad de 25 centímetros, esta medida no es muy relevante puesto que la trayectoria finaliza con la ascensión a este segundo escalón, lo único importante es que tenga una profundidad suficiente como para que toda la superficie de la planta del pie del robot pueda apoyarse en él.



(a) Vista lateral de las escaleras. Fuente: Elaboración propia



(b) Vista superior acotada de las escaleras. Fuente: Elaboración propia

Figura 4.10: Vistas de la escalera utilizada.

Este patrón de marcha es una marcha estática, el robot mantendrá el equilibrio siempre y cuando la proyección del centro de masas (F_{CoM}), se encuentre dentro del polígono de soporte, como ya se vio en el apartado 2.3.1 dedicado al criterio del ZMP, esta clase de movimientos solo son exitosos cuando las velocidades son muy bajas. Por lo tanto, el criterio principal a seguir para garantizar la estabilidad de la marcha será que la proyección en el suelo del centro de masas (F_{CoM}), permanezca siempre dentro del polígono de soporte establecido por el contacto de los pies del robot sobre el suelo.

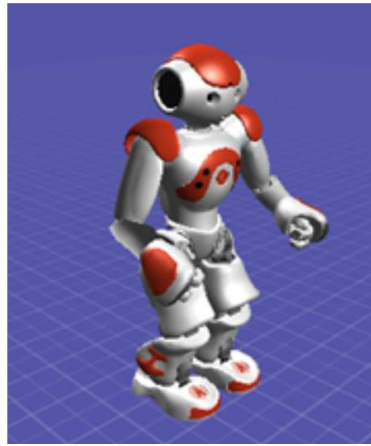
StandInit

Figura 4.11: Posición predefinida "StandInit". Fuente: http://doc.aldebaran.com/2-8/family/nao_technical/postures_naov6.html

El patrón de movimiento implementado se puede dividir en las siguientes etapas:

1. El robot parte de la posición inicial, predefinida por el fabricante, comienza un proceso de balanceo lateral hacia el lado izquierdo, hasta que la proyección centro de masas del robot (F_{CoM}), entra dentro del polígono de soporte del pie izquierdo, cuando se alcanza dicha posición el robot ya está preapropiado para elevar la pierna derecha sin que ello suponga una pérdida de equilibrio. A continuación, podemos ver en la figura 4.12 de forma simplificada, el patrón de movimiento mediante un modelo geométrico simplificado del robot NAO

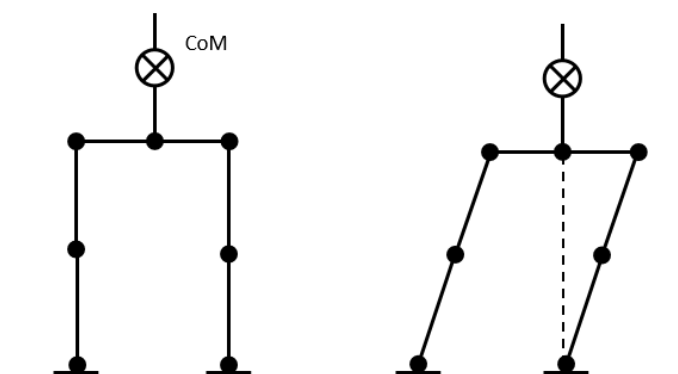
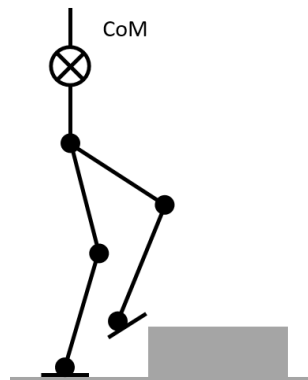
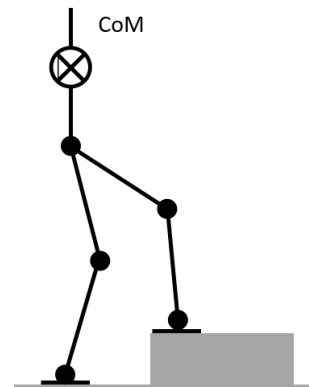


Figura 4.12: Movimiento 1º, movimiento lateral, de posición "StandInit" a posición de apoyo monopodal. Fuente: Elaboración propia

2. El pie derecho, una vez liberado de soportar el peso del robot (con el pie izquierdo actuando como pie de soporte), comienza a realizar su movimiento de oscilación, el movimiento debe elevar tanto el tobillo como la punta del pie de forma que el pie del robot no llegue a colisionar con el escalón, durante todo este proceso el FCoM permanece sobre el polígono de soporte, que, durante todo el periodo de oscilación, se sitúa debajo la planta del pie apoyado (pie izquierdo).



(a) Vista lateral del robot NAO justo cuando el pie derecho se encuentra realizando su movimiento oscilante. Fuente: Elaboración propia



(b) Vista lateral del robot NAO justo cuando el pie derecho acaba su movimiento oscilante y contacta con el escalón. Fuente: Elaboración propia

Figura 4.13: Croquis del movimiento 2

3. Una vez el pie derecho apoya la planta sobre el escalón se producen dos movimientos, que tienen como objetivo trasladar el FCoM de la planta del pie izquierdo a la planta del pie derecho en una trayectoria lo más rectilínea posible entre ambos puntos de forma que se garantice que siempre se encuentre dentro del polígono de soporte:
 - Movimiento frontal: Con el uso de ambas piernas se coordina un movimiento que desplaza el FCoM de atrás hacia delante.
 - Movimiento lateral: A la vez que el movimiento frontal, el robot inicia un movimiento de balanceo que desplaza el FCoM desde la izquierda hacia la derecha, quedándose dentro del polígono de soporte del pie derecho.

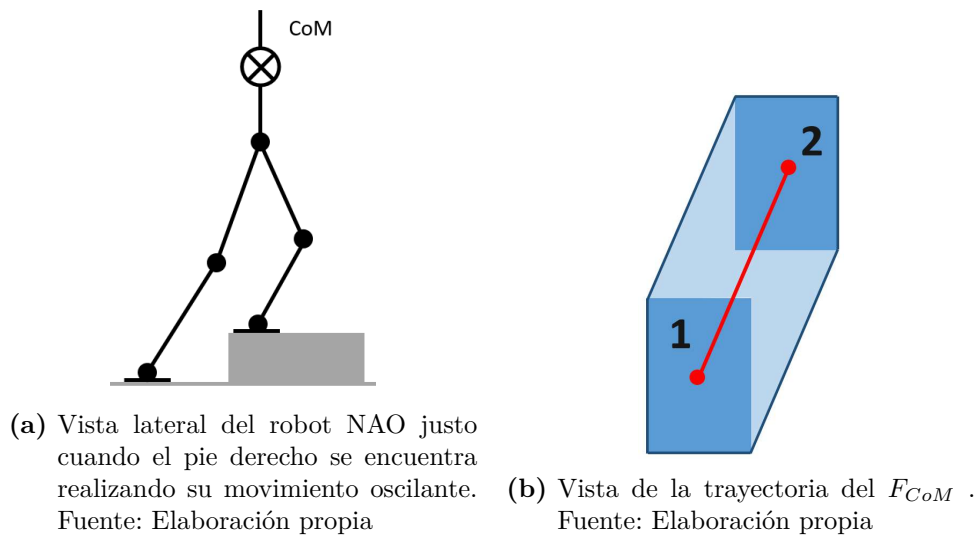


Figura 4.14: Croquis del movimiento 3

4. Una vez el FCoM se encuentra dentro del polígono de soporte, el robot puede elevar la pierna izquierda y subirla al escalón.

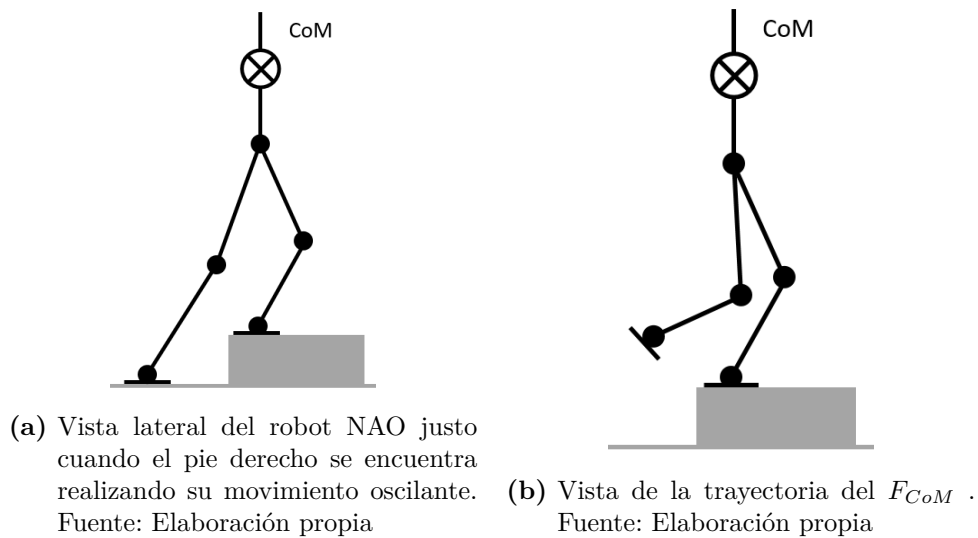
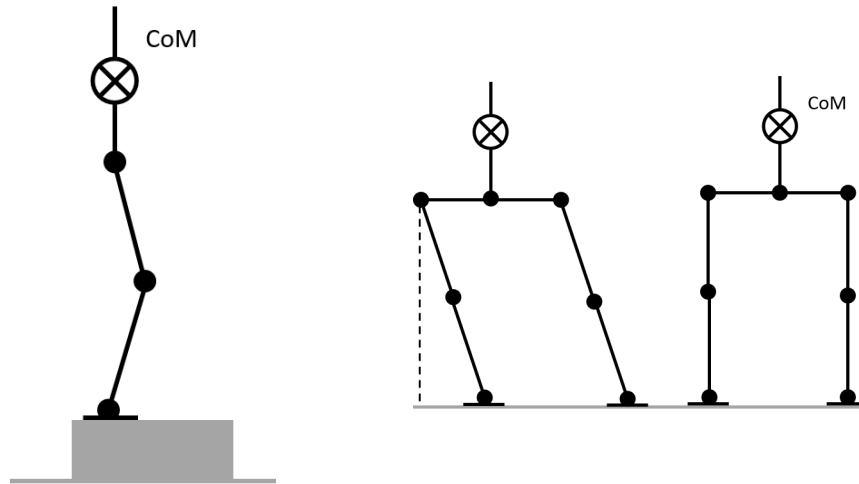


Figura 4.15: Croquis del movimiento 4

5. Con las dos piernas apoyadas en el escalón el robot procede a realizar un movimiento de oscilación que devuelva el FCoM a una posición intermedia y equidistante entre los dos pies, una vez recuperada la posición de apoyo doble, el robot vuelve a la posición inicial de la que partió.



(a) Vista lateral del robot NAO justo cuando el pie derecho se encuentra realizando su movimiento oscilante. (b) Vista de la trayectoria del F_{CoM} .
 Fuente: Elaboración propia Fuente: Elaboración propia

Figura 4.16: Croquis del movimientos 5

Una vez descrita la estrategia a implementar se aplica en el robot de una forma empírica, probando con diferentes patrones de movimiento para cada una de las articulaciones, analizando dichos movimientos desde el punto de vista de la estabilidad del robot a la hora de subir las escaleras. Aunque anteriormente se hayan enumerado por separado los diferentes movimientos que el robot debe realizar para subir las escaleras esto no quiere decir que cada uno de los movimientos sea temporalmente independiente del resto, por lo tanto, en la medida de lo posible, intentarán solapar unos con otros, lo que derivará en un patrón de movimiento más fluido, veloz y de apariencia más natural, perdiendo esa impresión de movimiento "robótico".

5 Código Implementado

A continuación, se describe el código utilizado para dotar al robot de la capacidad de realizar esta nueva trayectoria que le permita subir escaleras.

Primeramente, tenemos una línea de código donde el robot a través del código 5.1, avisa de que va a empezar a ejecutar del movimiento que le permite subir el primer escalón.

Código 5.1: Código Python para hablar, NAO

```
1 self.tts.say("Inicio")
```

En segundo lugar, se crean tres listas vacías con los tres parámetros que son necesarios definir; los nombres de las articulaciones a actuar, el tiempo en segundos en el que tienen que realizar cierta acción y la posición en radianes que tienen que alcanzar los motores de las articulaciones.

Código 5.2: Código Python listas vacías

```
1 names = list()
2 times = list()
3 keys = list()
```

A continuación se procede al llenado de las listas vacías. Con el método "append" añadimos la información necesaria a las listas vacías que hemos creado previamente, esta es la parte del código que añade la información de la trayectoria del movimiento.

Código 5.3: Ejemplo del código relativo únicamente a la articulación LKneePitch (flexión de la rodilla izquierda)

```
1 names.append("LKneePitch")
2 times.append([4.5, 8.5, 13.0, 17.0, 18.0, 19.0, 20.0, 24.0])
3 keys.append([0.0, 0.85, 0.25, 0.6, 0.6, 0.4, 2.0, 2.0])
```

Una vez definida la información del movimiento a realizar se llama al módulo `ALMotion`, el módulo de `NAOqi` encargado de las funciones, de control de fuerza y posición de las articulaciones, de las funciones de locomoción y del control Cartesiano. Dentro de este módulo, se incluye el método "angle Interpolation" que permite interpolar entre los diferentes objetivos o "targets" definidos en la lista `keys`.

Código 5.4: Código utilizado para llamar al módulo "ALMotion" y al método "angleInterepolation")

```
1 motion = ALProxy("ALMotion")
2 motion.angleInterpolation(names, keys, times, True)
```

Cuando usamos el módulo ALMotion debemos tener en cuenta el sistema de coordenadas que usa por defecto, (figura 5.1), además las unidades utilizadas serán las correspondientes al Sistema Internacional (SI), por lo tanto, éstas serán metros, segundos y radianes.

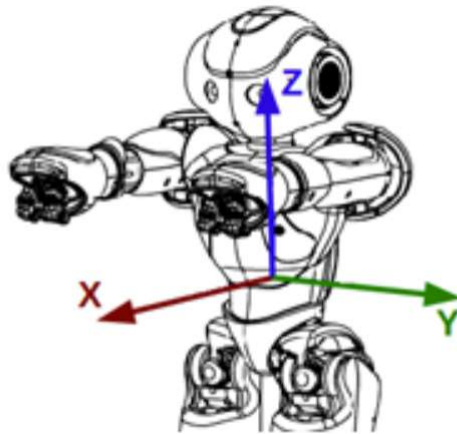


Figura 5.1: Sistema de referencia usado en el módulo ALMotion. Fuente: <http://doc.aldebaran.com/2-8/naoqi/motion/almotion.html>

Por último, cuando se ha llevado a cabo la acción que requeríamos se debe activar la salida del box para que se pueda activar la siguiente orden que se haya establecido en el programa.

Código 5.5: Código utilizado para llamar al módulo "ALMotion" y al método "angleInterepolation")

```
1 self.onStopped()
```


A continuación se adjunta el código completo implementado en ambas "box" de Choreagraphpe:

Código 5.6: Código en Python para el movimiento de subir escalón con la pierna derecha

```

1 class MyClass(GeneratedClass):
2     def __init__(self):
3         GeneratedClass.__init__(self)
4         self.tts = ALProxy('ALTextToSpeech')
5
6     def onLoad(self):
7         #put initialization code here
8         pass
9
10    def onUnload(self):
11        #put clean-up code here
12        pass
13
14    def onInput_onStart(self):
15
16        # Con el siguiente comando el robot dirá la palabra "Inicio",
17        # para avisarnos de que va a comenzar a realizar el movimiento
18
19        self.tts.say("Inicio")
20
21        # Creamos tres listas vacías con los tres parámetros que vamos a necesitar definir;
22        # los nombres de las articulaciones a actuar, el tiempo en segundos en el que tienen que realizar cierta acción,
23        # la posición en radianes que tienen que alcanzar los motores de las articulaciones.
24        names = list()
25        times = list()
26        keys = list()
27
28        # Con el método "append" añadimos la información necesaria a las listas vacías que hemos creado previamente,
29        # esta es la parte del código que añade la información de la trayectoria del movimiento.
30        names.append("RAnkleRoll")
31        times.append([2.5, 9.0, 15.0, 18.0, 20.0, 24.0, 26.0])
32        keys.append([0.15, 0.35, 0.1, -0.1, -0.21, -0.3, 0.0])
33
34        names.append("RHipRoll")
35        times.append([2.5, 9.0, 15.0, 18.0, 20.0, 24.0, 26.0])
36        keys.append([-0.15, -0.35, -0.1, -0.1, -0.21, -0.3, 0.0])
37
38        names.append("LAnkleRoll")
39        times.append([2.5, 9.0, 15.0, 18.0, 20.0, 24.0, 26.0])
40        keys.append([0.15, 0.35, 0.1, -0.1, -0.21, -0.3, 0.0])
41
42        names.append("LHipRoll")
43        times.append([2.5, 9.0, 15.0, 18.0, 20.0, 24.0, 26.0])
44        keys.append([-0.15, -0.35, -0.1, -0.1, -0.21, -0.3, 0.0])
45
46
47        names.append("LHipPitch")
48        times.append([4.5, 6.5, 13.0, 15.5, 18.0, 19.0, 23.0])
49        keys.append([0.0, -0.45, 0.3, 0.3, 0.9, 1.0, -0.75])
50
51        names.append("LKneePitch")
52        times.append([4.5, 8.5, 13.0, 17.0, 18.0, 19.0, 20.0, 24.0])
53        keys.append([0.0, 0.85, 0.25, 0.6, 0.6, 0.4, 2.0, 2.0])
54
55        names.append("LAnklePitch")
56        times.append([4.5, 7.0, 10.5, 14.0, 18.0, 19.0, 21.5, 24.0])
57        keys.append([0.0, -0.35, -0.6, -0.5, -0.95, -1.0, -1.95, -1.35])

```

```

58
59     names.append("RHipPitch")
60     times.append([2.5, 4.5, 10.5, 13.25, 18.0, 21.0, 22.0, 24.0])
61     keys.append([0.0, -0.45, -1.25, -1, -1.25, -1.25, -1.2, -0.75])
62
63     names.append("RKneePitch")
64     times.append([2.5, 4.5, 9.65, 13.25, 21.0, 24.0])
65     keys.append([0.0, 0.80, 1.7, 1.0, 2.2, 2.15])
66
67     names.append("RAnklePitch")
68     times.append([2.5, 4.5, 8.5, 15.5, 20.5, 24.0])
69     keys.append([0.0, -0.3, -0.6, 0.2, -0.95, -1.35])
70
71
72     names.append("LShoulderRoll")
73     times.append([1.0, 26.0])
74     keys.append([0.2, 0.2])
75
76     names.append("RShoulderRoll")
77     times.append([1.0, 26.0])
78     keys.append([-0.2, -0.2])
79
80
81     try:
82         # uncomment the following line and modify the IP if you use this script outside Choregraphe.
83         # motion = ALProxy("ALMotion", IP, 9559)
84
85         # Llamamos al módulo AlMotion, el módulo de Naoqi encargado de las funciones, de control de fuerza
86         # y posición de las articulaciones, de las funciones de locomoción y del control Cartesiano
87         motion = ALProxy("ALMotion")
88         # Usamos el método "angle Interpolation" para interpolar entre los diferentes obetivos o "targets"
89         # definidos en la lista keys.
90         motion.angleInterpolation(names, keys, times, True)
91         # Una vez termina el movimiento activamos la salida del box para que se produzca la siguiente acción
92         self.onStopped()
93     except BaseException, err:
94         print err
95
96     pass
97
98 def onInput_onStop(self):
99     self.onUnload() #it is recommended to reuse the clean-up as the box is stopped
100    self.onStopped() #activate the output of the box

```

Código 5.7: Código en Python para el movimiento de subir escalón con la pierna izquierda

```

1 class MyClass(GeneratedClass):
2     def __init__(self):
3         GeneratedClass.__init__(self)
4         self.tts = ALProxy('ALTextToSpeech')
5
6     def onLoad(self):
7         #put initialization code here
8         pass
9
10    def onUnload(self):
11        #put clean-up code here
12        pass
13
14    def onInput_onStart(self):
15
16        # Con el siguiente comando el robot dirá "Movimiento 2",
17        # para avisarnos de que va a comenzar a realizar el segundo movimiento,
18        # que será el que utilizaremos para subir el segundo escalón con la pierna izquierda
19        self.tts.say("Movimiento dos")
20
21        # Creamos tres listas vacías con los tres parámetros que vamos a necesitar definir;
22        # los nombres de las articulaciones a actuar, el tiempo en segundos en el que tienen que realizar cierta acción,
23        # la posición en radianes que tienen que alcanzar los motores de las articulaciones.
24        names = list()
25        times = list()
26        keys = list()
27
28        # Con el método "append" añadimos la información necesaria a las listas vacías que hemos creado previamente,
29        # esta es la parte del código que añade la información de la trayectoria del movimiento.
30        names.append("RAnkleRoll")
31        times.append([2.5, 9.0, 10.5, 15.0, 18.0, 20.0, 24.0, 26.0])
32        keys.append([-0.15, -0.32, -0.32, -0.1, 0.1, 0.21, 0.3, 0.0])
33
34        names.append("RHipRoll")
35        times.append([2.5, 9.0, 10.5, 15.0, 18.0, 20.0, 24.0, 26.0])
36        keys.append([0.15, 0.32, 0.32, 0.1, 0.1, 0.21, 0.3, 0.0])
37
38        names.append("LAnkleRoll")
39        times.append([2.5, 9.0, 10.5, 15.0, 18.0, 20.0, 24.0, 26.0])
40        keys.append([-0.15, -0.32, -0.32, -0.1, 0.1, 0.21, 0.3, 0.0])
41
42        names.append("LHipRoll")
43        times.append([2.5, 9.0, 10.5, 15.0, 18.0, 20.0, 24.0, 26.0])
44        keys.append([0.15, 0.32, 0.32, 0.1, 0.1, 0.21, 0.3, 0.0])
45
46
47        names.append("RHipPitch")
48        times.append([4.5, 6.5, 13.0, 15.5, 18.0, 19.0, 23.0])
49        keys.append([0.0, -0.45, 0.3, 0.3, 0.9, 1.0, -0.75])
50
51        names.append("RKneePitch")
52        times.append([4.5, 8.5, 13.0, 17.0, 18.0, 19.0, 20.0, 24.0])
53        keys.append([0.0, 0.85, 0.25, 0.6, 0.6, 0.4, 2.0, 2.0])
54
55        names.append("RAnklePitch")
56        times.append([4.5, 7.0, 10.5, 14.0, 18.0, 19.0, 21.5, 24.0])
57        keys.append([0.0, -0.35, -0.6, -0.5, -0.95, -1.0, -1.95, -1.35])
58
59        names.append("LHipPitch")

```

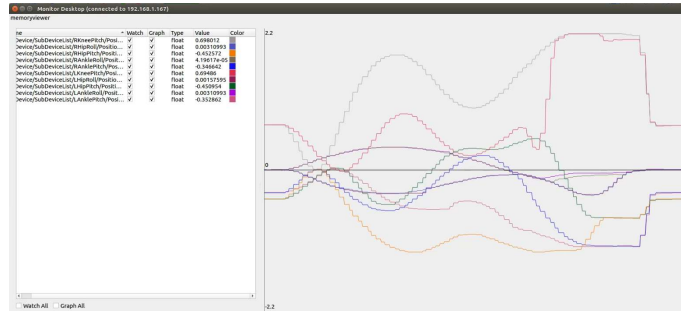
```

60     times.append([2.5, 4.5, 10.5, 13.25, 18.0, 21.0, 22.0, 24.0])
61     keys.append([0.0, -0.45, -1.25, -1, -1.25, -1.25, -1.2, -0.75])
62
63     names.append("LKneePitch")
64     times.append([2.5, 4.5, 9.65, 13.25, 21.0, 24.0])
65     keys.append([0.0, 0.80, 1.7, 1.0, 2.2, 2.15])
66
67     names.append("LAnklePitch")
68     times.append([2.5, 4.5, 8.5, 15.5, 20.5, 24.0])
69     keys.append([0.0, -0.4, -0.6, 0.2, -0.95, -1.35])
70
71
72     names.append("LShoulderRoll")
73     times.append([2.0, 26.0])
74     keys.append([0.2, 0.2])
75
76     names.append("RShoulderRoll")
77     times.append([2.0, 26.0])
78     keys.append([-0.2, -0.2])
79
80
81     try:
82         # uncomment the following line and modify the IP if you use this script outside Choregraphe.
83         # motion = ALProxy("ALMotion", IP, 9559)
84
85         # Llamamos al módulo ALMotion, el módulo de Naoqi encargado de las funciones, de control de fuerza
86         # y posición de las articulaciones, de las funciones de locomoción y del control Cartesiano
87         motion = ALProxy("ALMotion")
88         # Usamos el método "angle Interpolation" para interpolar entre los diferentes objetivos o "targets"
89         # definidos en la lista keys.
90         motion.angleInterpolation(names, keys, times, True)
91         # Una vez termina el movimiento activamos la salida del box para que se produzca la siguiente acción
92         self.onStopped()
93
94     except BaseException, err:
95         print err
96
97     pass
98
99     def onInput_onStop(self):
100         self.onUnload() #it is recommended to reuse the clean-up as the box is stopped
101         self.onStopped() #activate the output of the box

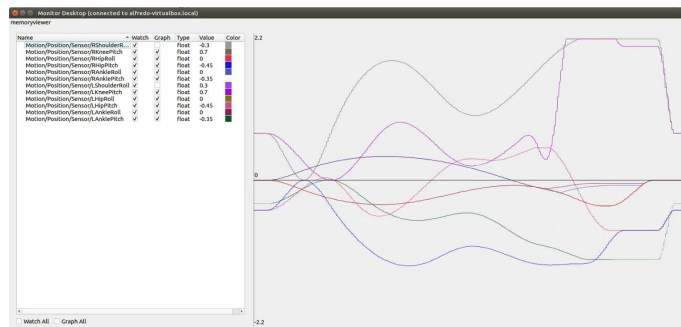
```

6 Resultados

El objetivo principal del trabajo era desarrollar un patrón de marcha capaz de conseguir que el robot NAO subiese escaleras. En primer lugar se planificó una estrategia general que asegurase que el F_{CoM} se mantuviese siempre dentro del polígono de soporte del robot. Seguidamente se aplicó esta estrategia de forma empírica, analizando la influencia de los movimientos de cada una de las articulaciones a la estabilidad del robot. El resultado ha sido que el robot es capaz de subir un escalón de 2.8 centímetros en un tiempo total de 26 segundos. Los movimientos que realiza cada articulación están definidos en las siguientes gráficas:

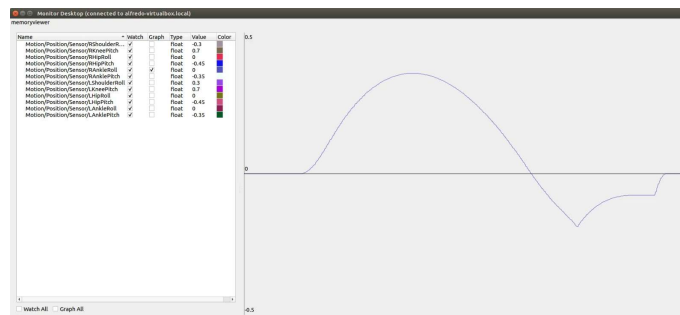


(a) Evolución de la posición angular con respecto al tiempo de todas las articulaciones implicadas en el movimiento

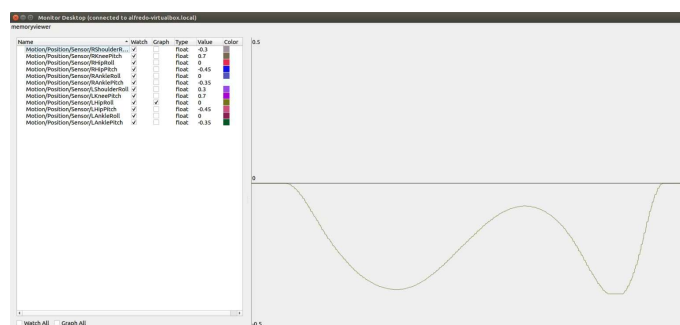


(b) Evolución de la posición angular con respecto al tiempo del movimiento de flexión de la rodilla derecha

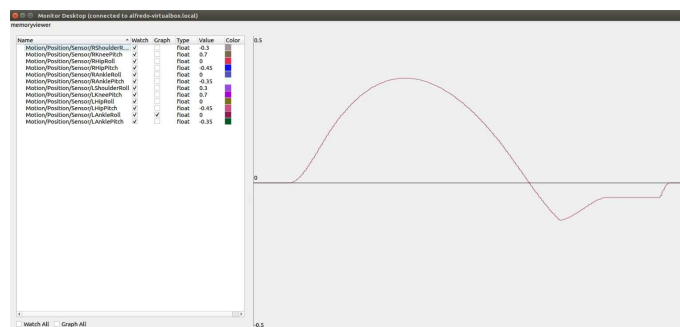
Figura 6.1: Evolución de la posición angular con respecto del tiempo en las articulaciones del robot



(f) Evolución de la posición angular con respecto al tiempo del movimiento de abducción del tobillo derecho

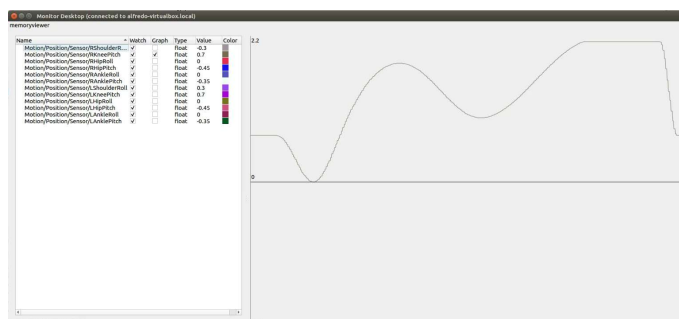


(g) Evolución de la posición angular con respecto al tiempo del movimiento de abducción de la cadera izquierda

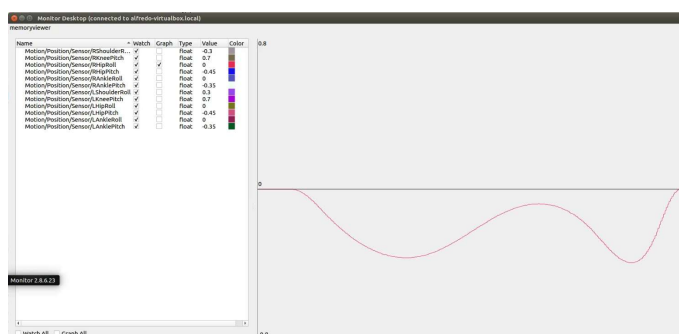


(h) Evolución de la posición angular con respecto al tiempo del movimiento de abducción del tobillo izquierdo

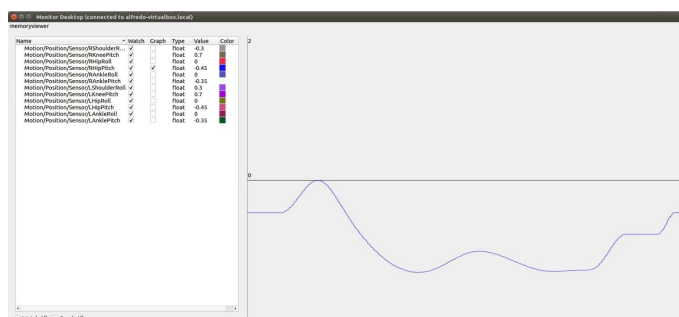
Figura 6.1: Evolución de la posición angular con respecto del tiempo en las articulaciones del robot



(c) Evolución de la posición angular con respecto al tiempo del movimiento de flexión de la rodilla derecha

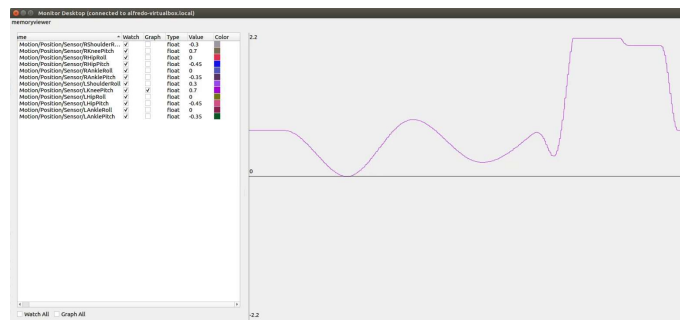


(d) Evolución de la posición angular con respecto al tiempo del movimiento de abducción de la cadera derecha

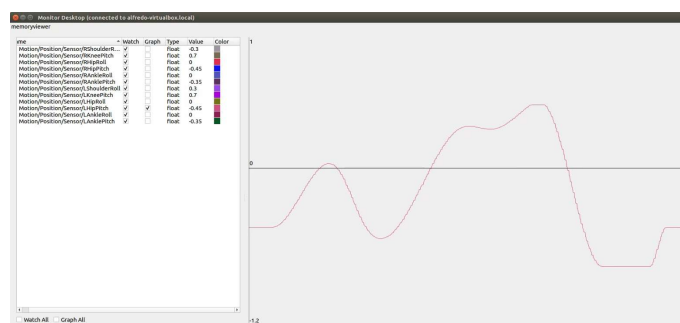


(e) Evolución de la posición angular con respecto al tiempo del movimiento de flexión de la cadera derecha

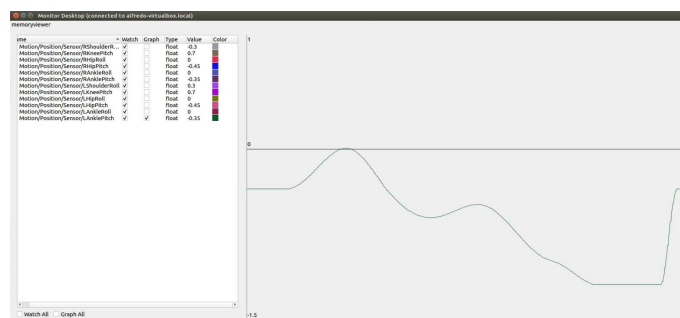
Figura 6.1: Evolución de la posición angular con respecto del tiempo en las articulaciones del robot



(i) Evolución de la posición angular con respecto al tiempo del movimiento de flexión de la rodilla izquierdo



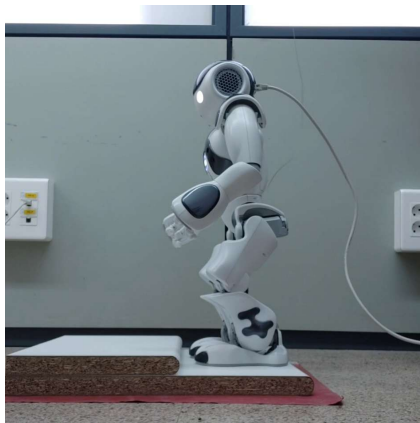
(j) Evolución de la posición angular con respecto al tiempo del movimiento de flexión de la cadera izquierda



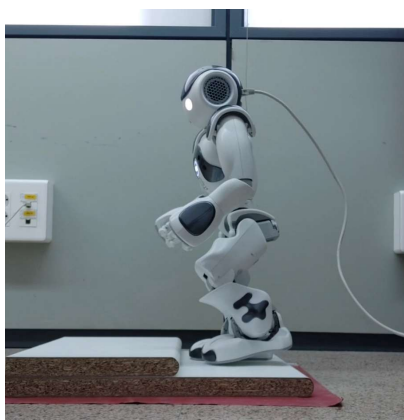
(k) Evolución de la posición angular con respecto al tiempo del movimiento de flexión del tobillo izquierdo

Figura 6.1: Evolución de la posición angular con respecto del tiempo en las articulaciones del robot

En las siguientes imágenes se puede ver al robot NAO realizando la trayectoria para subir las escaleras. En las imágenes se muestran los momentos más importantes de la trayectoria cuando el robot utiliza la pierna izquierda para subir el escalón:

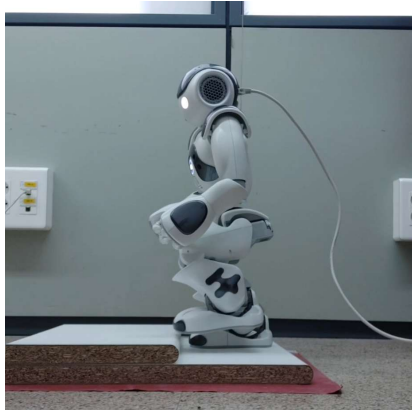


(a) Posición inicial del Robot en su posición biapoyada. $t=0s$



(b) El robot levanta la pierna izquierda y empieza la fase de soporte de apoyo simple. $t=5.0s$

Figura 6.2: Imágenes del robot NAO subiendo las escaleras, momentos destacados del ciclo



(c) El pie oscilante sobrepasa la altura del escalón. $t=10.0s$



(d) El pie izquierdo contacta con el escalón utilizando solo el antepie. $t=13.0s$



(e) El pie izquierdo se encuentra en contacto total con el escalón. $t=17.0s$

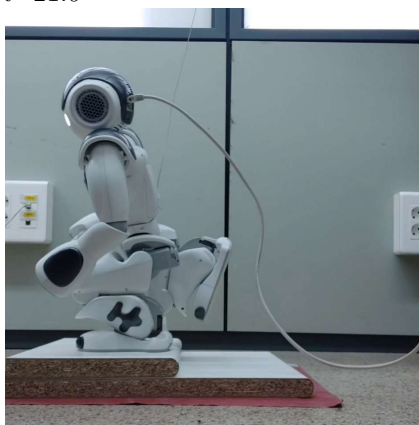
Figura 6.2: Imagenes del robot NAO subiendo las escaleras, momentos destacados del ciclo



(f) El pie derecho se empieza a elevar mientras el centro de gravedad del robot se desplaza hacia delante. $t=20.0s$

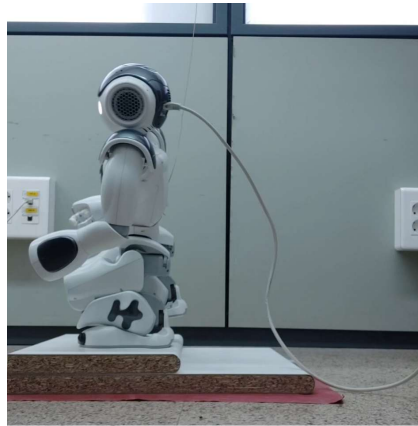


(g) El pie derecho se despega totalmente del suelo. $t=21.0$

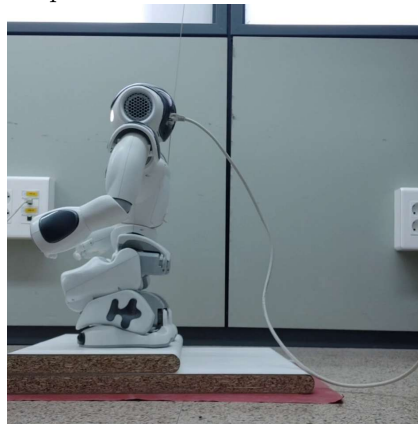


(h) La rodilla derecha se flexiona rápidamente para evitar el escalón. $t=22.0s$

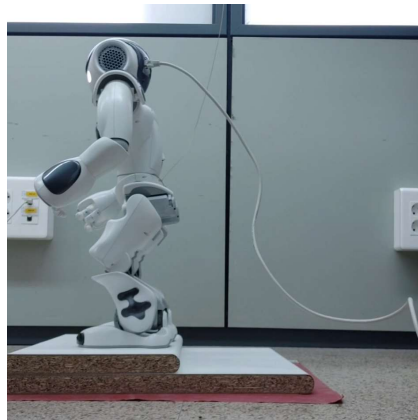
Figura 6.2: Imágenes del robot NAO subiendo las escaleras, momentos destacados del ciclo



- (i) La cadera derecha desplaza la pierna hacia delante para colocarla sobre el escalón $t=24.0$



- (j) El pie derecho contacta con el escalón. $t=26.0s$



- (k) El robot recupera la posición inicial. $t=28.0s$

Figura 6.2: Imagenes del robot NAO subiendo las escaleras, momentos destacados del ciclo

Para realizar una secuencia en la que el robot NAO suba unas escaleras de dos escalones, cada uno de ellos con una pierna diferente se genera el programa en Choregraphe que se puede ver en la figura 6.3. Donde podemos ver en el Workflow de Choregraphe 5 bloques conectados en serie, tres de ellos (el primero, el tercero y el quinto), sirven para indicar al robot que vuelva a la posición erguida, estos bloques vienen predefinidos en el propio programa. Mientras que los otros dos bloques, son los bloques que se han implementado en este proyecto, cada uno de ellos sirve para subir el escalón con una pierna distinta.

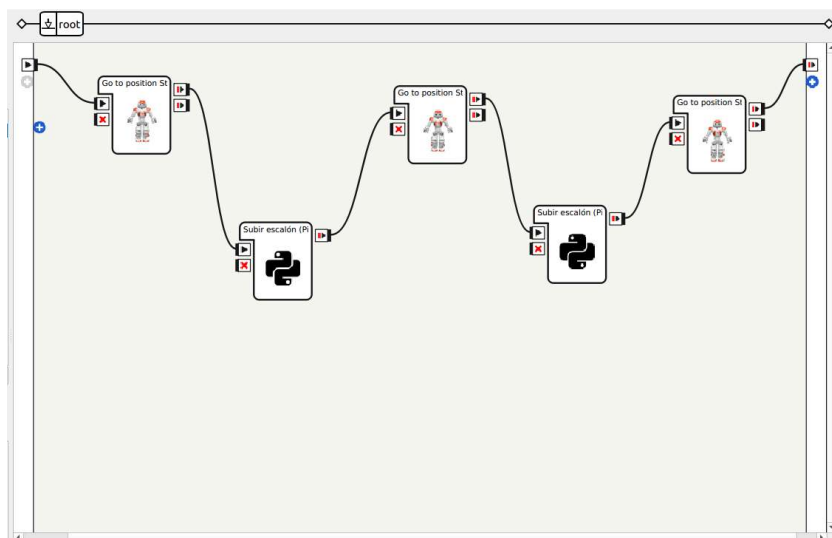


Figura 6.3: Workflow de Choregraphe con el programa cargado. Fuente: Elaboración propia

7 Conclusiones

Una vez expuesto el trabajo realizado, que ha abarcado desde el estudio biomecánico de la marcha humana, pasando por los algoritmos más empleados en la locomoción de robots humanoides y demás estrategias de locomoción aplicadas en robots humanoides bípedos, llega la hora de aportar las conclusiones obtenidas de nuestra investigación y experimentación con los patrones de marcha humanoides para subir escaleras con el robot NAO.

En primer lugar, cabe destacar la importancia y vigencia de todos los conocimientos expuestos, en especial el estudio del método ZMP, en el que se basan gran parte de los sistemas de locomoción bípeda para robots humanoides actuales.

En cuanto al diseño de la trayectoria elaborada para el robot NAO, cabe destacar la gran dificultad que supone desarrollar este tipo de trayectorias sin recurrir a un algoritmo (ya sea basado en ZMP u otro método) que sea el encargado de definir las trayectorias articulares a seguir en base a las condiciones de equilibrio establecidas, de esta forma el robot no siempre realizará la misma trayectoria, si no que irá adecuando la trayectoria a las condiciones establecidas en el algoritmo.

Este es el principal reto que nos encontramos, la imposibilidad de que el robot se pueda adaptar de algún modo a las condiciones del entorno, ya que los patrones de movimiento articular vienen ya preestablecidos en base a los ensayos empíricos realizados. Por lo tanto, cualquier variación en las condiciones en las que se realice la subida del escalón, como podrían ser las imperfecciones del entorno (suelo no uniforme con inclinaciones, algún tipo de defecto en el escalón, variación en la altura del escalón...) podrían causar alteraciones fatales en el posicionamiento del centro de gravedad del robot con respecto al polígono de soporte, de tal modo que el robot no podrá mantener el equilibrio.

Por lo tanto, la conclusión a la que podemos llegar tras el desarrollo de este trabajo es que, cuando se realizan patrones de movimientos tan complejos, y que requieren de mucha precisión para que el patrón de movimiento alcance el objetivo deseado, aunque es posible realizarlo de una forma empírica, siempre desarrollando antes unos conocimientos teóricos que guíen nuestros experimentos, como hemos visto en el apartado 4.3, donde se establecía una estrategia de movimiento en base a la posición que debe tener el centro de gravedad del robot para alcanzar un resultado exitoso, no es lo más recomendable, porque, aunque los resultados obtenidos, en situaciones ideales, pueden ser igual de buenos, incluso mejores, que los que se pueden conseguir implementando un algoritmo de planeación de trayectorias, nuestro método de subida de escalones nunca se podrá adaptar a diferentes situaciones.

7.1 Trabajos futuros

En base a las conclusiones obtenidas en este proyecto, se proponen ciertas mejoras que se pueden realizar, partiendo de los conocimientos obtenidos gracias a la experimentación en este proyecto, para poder llevar a cabo en trabajos futuros:

- Implementación de un sistema de visión artificial, capaz de reconocer los escalones y poder medir la distancia que hay entre un escalón y otro, de forma que, si la distancia entre escalones es más amplia que los escalones utilizados en este proyecto, el robot sea capaz de calcular que distancia debe recorrer, caminando, para posicionarse de forma óptima con respecto al escalón.
 - Basándose en los conocimientos obtenidos en este proyecto con respecto a las estrategias para abordar la subida de escalones, implementar un algoritmo de planificación de trayectorias que permita al robot NAO subir escalones, sustituyendo así a las trayectorias articulares que se han programado en este proyecto.
-

Bibliografía

Asimo. (Fecha de la última visita: 02/06/21). <https://es.wikipedia.org/wiki/ASIMO>.

Asimo el robot humanoide más avanzado del mundo. (Fecha de la última visita: 02/06/21). <https://www.honda.mx/asimo>.

Atlas. (Fecha de la última visita: 05/06/21). [https://es.wikipedia.org/wiki/Atlas_\(robot\)](https://es.wikipedia.org/wiki/Atlas_(robot)).

Biomecánica de la marcha y la carrera. (Fecha de la última visita: 08/06/21). https://www.munideporte.com/imagenes/documentacion/ficheros/20091229110320blanca_de_la_cruz2.pdf.

Biomedical analysis of body mass transfer during stair ascent and descent of healthy subjects. (1993). *The Journal of Rehabilitation Research and Development*. https://www.researchgate.net/profile/James-Zachazewski/publication/15039814_Biomedical_analysis_of_body_mass_transfer_during_stair_ascent_and_descent_of_healthy_subjects/links/0912f511a9dcce178b000000/Biomedical-analysis-of-body-mass-transfer-during-stair-ascent-and-descent-of-healthy-subjects.pdf.

Cinesiología de la marcha humana normal. (Fecha de la última visita: 08/06/21). <http://wzar.unizar.es/acad/cinesio/Documentos/Marcha20humana.pdf>.

Dekker, M. (2009). Zero-moment point method for stable biped walking. https://www.techunited.nl/media/files/humanoid/MaartenDekker_OPEN2009_Zero_Moment_Point_Method_for_Stable_Biped_Walking.pdf.

Documentación choregraphe. (Fecha de la última visita: 08/06/21). http://doc.aldebaran.com/2-8/news/2.8/choregraphe_rn2.8.html.

Documentación monitor 2.8. (Fecha de la última visita: 08/06/21). <http://doc.aldebaran.com/2-8/software/monitor/index.html>.

Documentación nao 2.8. (Fecha de la última visita: 03/07/21). http://doc.aldebaran.com/2-8/home_nao.html.

Documentación naoqi 2.8. (Fecha de la última visita: 03/07/21). http://doc.aldebaran.com/2-8/index_dev_guide.html.

Doríforo. (Fecha de la última visita: 14/06/21). <https://es.wikipedia.org/wiki/Doríforo>.

- Dynamics, B. (Fecha de la última visita: 05/06/21). *Atlas*. <https://www.bostondynamics.com/atlas>.
- Efficient bipedal robots based on passive-dynamic walkers*. (2005). http://ruina.tam.cornell.edu/research/topics/locomotion_and_robotics/efficient_bipedal_robots/efficient_bipedal_robots.pdf.
- El doríforo de policleto, la escultura que marcó el canon*. (Fecha de la última visita: 14/06/21). <https://grupenciclopedia.cat/blog/es/el-doriforo-de-policleto/>.
- Fases de la marcha humana. (s.f.). <https://www.elsevier.es/es-revista-revista-iberoamericana-fisioterapia-kinesiologia-176-articulo-fases-marcha-humana-13012714>.
- Hombre de vitrubio*. (Fecha de la última visita: 06/06/21). https://es.wikipedia.org/wiki/Hombre_de_Vitruvio.
- Humanoid Robotics Institute, W. U. (Fecha de la última visita: 07/06/21). *Wabot waseda robot*. https://www.humanoid.waseda.ac.jp/booklet/kato_2.html.
- Miomir Vukobratovic, B. B. (2004). Zero-moment point - thirty years of its life. <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.914.2204rep=rep1type=pdf>.
- Nao, los robots del futuro son ya una realidad*. (Fecha de la última visita: 04/06/21). <https://aliverobots.com/nao/>.
- Nao v6*. (Fecha de la última visita: 04/06/21). <https://www.softbankrobotics.com/emea/es/nao>.
- Oort, G. V., y Stramigioli, S. (1996). Geometric interpretation of the zero-moment point. <https://core.ac.uk/download/pdf/11479356.pdf>.
- Protocolo de evaluación de un sistema para medición de parámetros de tiempo de la marcha humana. (2017). https://www.researchgate.net/publication/321492779_Protocolo_de_Evaluacion_de_un_Sistema_para_Medicion_de_Parametros_de_Tiempo_de_la_Marcha_Humana.
- Punto de momento cero*. (Fecha de la última visita: 02/06/21). https://es.wikipedia.org/wiki/Punto_de_Momento_Cero.
- Robocup*. (Fecha de la última visita: 04/06/21). <https://es.wikipedia.org/wiki/Robocup>.
- Robocupsoccer - humanoid*. (Fecha de la última visita: 04/06/21). <https://www.robocup.org/leagues/3>.
- Robot de leonardo*. (Fecha de la última visita: 14/06/21). https://es.wikipedia.org/wiki/Robot_de_Leonardo.
- Stair ascent and descent at different inclinations. (2001). <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.534.3478&rep=rep1&type=pdf>.
-

Stairclimbing kinematics on stairs of differing dimensions. (1991). [https://www.archives-pmr.org/article/0003-9993\(91\)90174-H/pdf](https://www.archives-pmr.org/article/0003-9993(91)90174-H/pdf).

Síntesis de la marcha robótica mediante la aproximación del ciclo límite: El robot bípedo esbirro. (Fecha de la última visita: 08/06/21). <https://intranet.ceautomatica.es/old/actividades/jornadas/XXIX/pdf/240.pdf>.

Target location and gait planning for humanoid robot climbing stairs. (2019). *2019 IEEE 4th International Conference on Advanced Robotics and Mechatronics (ICARM)*.

Wabot, la familia de robots humanoides de la universidad de waseda. (Fecha de la última visita: 19/06/21). <https://www.abadiadigital.com/wabot-la-familia-de-robots-humanoides-de-la-universidad-de-waseda/>.

Lista de Acrónimos y Abreviaturas

CdG	Centro de Gravedad.
CoM	Center of Mass.
CoP	Center of Pressure.
MIT	Massachusetts Institute of Technology.
SP	Polígono de Soporte.
TFM	Trabajo Final de Máster.
ZMP	Zero-Moment Point.